DOCUMENT RESUME

ED 178 321 SE 029 202

| | |
|---|---|
| AUTHOR | Reuss, E.; And Others |
| TITLE | FORTRAN IV Digital Filter Design Programs. Digital Systems Education Project. |
| INSTITUTION | Pittsburgh Univ., Pa. Dept. of Electrical Engineering. |
| SPONS AGENCY | National Science Foundation, Washington, D.C. |
| PUB DATE | 76 |
| GRANT | NSF-GZ-2997 |
| NOTE | 94p. |
| EDRS PRICE | MF01/PC04 Plus Postage. |
| DESCRIPTORS | Computer Oriented Programs; *Computer Programs; *Digital Computers; Educational Technology; *Instructional Materials; Programing; *Technical Reports |
| IDENTIFIERS | *Digital Filter |

ABSTRACT

The goals of the Digital Systems Education Project (DISE) include the development and distribution of educational/instructional materials in the digital systems area. Toward that end, this document contains three reports: (1) A FORTRAN IV Design Program for Low-Pass Butterworth and Chebychev Digital Filters; (2) A FORTRAN IV Design Program for Butterworth and Chebychev Band-Pass and Band-Stop Digital Filters; and (3) Programs for Weighted Least Squares Design of Nonrecursive and Recursive Digital Filters. The first two reports give the design procedure used, a description of the program, and design examples for the respective programs. The third report describes the operation of the two programs under discussion and gives examples to illustrate their operation. (MK)

DISE

DTS-01

ED178321

# FORTRAN IV DIGITAL FILTER DESIGN PROGRAMS

E. Reuss H.J. Markos F.L. Mann T.A. Brubaker

ELECTRICAL ENGINEERING DEPARTMENT
COLORADO STATE UNIVERSITY
FORT COLLINS
COLORADO

DIGITAL SYSTEMS EDUCATION COMMITTEE
DISCRETE TIME SYSTEMS TASK FORCE



DISE is a project supported by the National Science Foundation
Grant No. GZ-2997

# DISE PROJECT

The DISE (Digital Systems Education) Project is sponsored by Grant #GZ-2997 from the National Science Foundation. The nucleus of the project is the DISE Advisory Committee, which is an inter-university, inter-industry working group with the specific charter of developing, coordinating the development of, and distributing educational/instructional materials in the digital systems area.

The specific goals of the project are: to assess Digital Systems Education, both to determine the types of curricula, course contents, lab structures, etc., in present programs and to determine how present programs are meeting the needs of industry and the students; to review existing educational/instructional materials in this area; to develop and/or coordinate the development of new materials; to provide a industry/university forum to foster the exchange of new technology; and to obtain widespread dissemination and use of newly developed or existing materials.

## Project Structure

Advisory Committee. It is the responsibility of this group to determine the areas in which the educational/instructional materials will be solicited and developed and to act as a review board for proposed projects and completed materials. The committee will consist of between 15 and 20 members with the academic, industrial, and professional societies sectors represented. The present members of this committee are:

Dr. Wayne Black
Charles T. Main, Inc.

Professor Taylor Booth
University of Connecticut

Professor Thomas Brubaker
Colorado State University

Professor James T. Cain
University of Pittsburgh

Professor Yaohan Chu
University of Maryland

Professor Ben Coates
Purdue University

Professor Ronald G. Hoelzeman
University of Pittsburgh

Dr. Glen Langdon
IBM Research

Professor Arthur Lo
Princeton University

Mr. Francis Lynch
National Semiconductor Corporation

Professor Larry McNamee
UCLA

Professor T.W. Sze
University of Pittsburgh

Professor H.C. Torng
Cornell University

Professor Raymond Voith
University of Toledo

Task Force Committees (Project Groups). These committees or groups represent the "manufacturing" division of DISE and constitute the working groups of people developing materials for dissemination. There are presently five project groups, although new groups will be formed due to changes in technology, demand, or interest levels.

For further information, contact:

DISE
Electrical Engineering Department
University of Pittsburgh
Pittsburgh, Pennsylvania 15261

# A FORTRAN IV DESIGN PROGRAM FOR LOW-PASS BUTTERWORTH AND CHEBYCHEV DIGITAL FILTERS

by

H. J. Markos

and

T. A. Brubaker

The authors are with the Electrical Engineering Department at Colorado State University, Fort Collins, Colorado

# INTRODUCTION

This report contains the documentation for the LPASS program. It consists of the design procedure used, a description of the program, and design examples using the program.

The purpose of the LPASS program is the design of a maximally flat Butterworth or an equiripple Chebychev lowpass digital filter. Starting with an analog filter, the bilinear Z transform is used to find an equivalent digital filter. The user enters the following parameters: the number of second order sections, the type of filter, the sampling interval, the -3db cutoff frequency, the starting frequency and the frequency increment. If a Chebychev filter is being designed, the ripple must also be entered.

The program calculates the digital filter coefficients for up to three second order sections in cascade. The program is designed to calculate up to a sixth order filter, thus the filter order is two times the number of cascaded second order sections. The filter magnitude response is generated over the frequency interval specified by the input.

The LPASS program, written in Fortran IV, is supplied as a card deck with this report. The program is in the form of a subroutine and can be used as is by a call statement from the main program. Data may be input via cards with output available through a line printer. The input/output devices may be altered as explained in this report. Graphics routines may easily be appended to the program.

## I. Design Procedure

### A. Preliminary Discussion

The transfer function of a second order digital filter in the Z domain is given by

$$H(Z) = \frac{K_1(A_0Z^2+A_1Z+A_2)}{Z^2+B_1Z+B_2} \tag{1}$$

where the A's and B's are the coefficients of the numerator and denominator respectively. One common method of designing a digital filter is to start with an analog transfer function H(S) and transform it to the digital transfer function H(Z). This program will calculate the scale factor $K_1$ and the coefficients $A_0$, $A_1$, $A_2$, $B_1$, and $B_2$. The transformation used is the extended bilinear Z transform defined as

$$S \rightarrow \frac{2}{T}\left(\frac{Z-1}{Z+1}\right), \tag{2}$$

where T is the sampling interval. When this transform is employed, the desired frequencies must first be prewarped to make them compatible with the digital filter. The prewarped cutoff frequency is given by

$$WDC = \frac{2}{T}\,\mathrm{Tan}\left(\frac{\omega_c T}{2}\right). \tag{3}$$

This prewarping is done by the program.

### B. Butterworth Low-Pass Filter

We start with a normalized second order low-pass filter in the S plane.

$$H(S) = \frac{1}{S^2 + 2S\cos\theta + 1} \tag{4}$$

where the angle $\theta$ is in degrees (in the program). $\theta$ may be found from the Butterworth circle and the relationship

$$S = e^{\pm j\pi(2m-1)/2n} \tag{5}$$

where $n$ is the order of the filter and $m = 1, 2, 3, \ldots, n$. This relationship is determined by the following procedure. By definition, a filter is $n^{th}$ order Butterworth low-pass if its gain characteristic is

$$|H_n(j\omega)|^2 = \frac{a^2}{1 + (\frac{\omega}{\omega_c})^{2n}} \tag{6}$$

where $a$ is the gain, $\omega_c$ is the desired cutoff frequency and $n$ is the order of the filter. Note that $|H_n(j\omega)|^2$ goes to zero as $\omega$ goes to infinity, indicating the filter does attenuate the higher frequencies. To determine its efficiency as a low-pass filter we calculate

$$\frac{d}{d\omega}|H_n(j\omega)| = -\frac{an}{\omega_c} \frac{(\frac{\omega}{\omega_c})^{2n-1}}{\left[1+(\frac{\omega}{\omega_c})^{2n}\right]^{3/2}} \tag{7}$$

Thus

$$\frac{d}{d\omega}\left[|H_n(j\omega)|\right]_{\omega=0} = 0 \tag{8}$$

for all $n$ and hence the gain characteristic stays flat for $\omega$ close to 0. Also

$$\left[\frac{d}{d\omega}|H_n(j\omega)|\right]_{\omega=\omega_c} = -\frac{an}{2\omega_c\sqrt{2}} \tag{9}$$

and hence, the decline rate or "roll-off" of the gain characteristic

at $\omega = \omega_c$ becomes sharper as n increases. In other words, the approximation to the ideal low-pass filter improves for larger n. The order n is chosen according to desired specifications. The references have equations, curves, and tables that select n, given the specifications. For example, page 227 of Rabiner and Gold gives an equation for calculating n when the transition band is specified.

In the design, the poles for the full frequency response, H(S), of the $n^{th}$ order Butterworth filter must be determined. The procedure is as follows:

$$|H_n(j\omega)|^2 = H_n(j\omega)\overline{H_n(j\omega)} = H_n(j\omega)H_n\overline{(j\omega)} = H_n(j\omega)H_n(-j\omega)$$

$$= [H(S)H(-S)]_{S=j\omega} = \left[\frac{a^2}{1 + (\frac{\omega}{\omega_c})^{2n}}\right]_{\omega = \frac{S}{j}} = \frac{a^2}{1 + (\frac{S}{j\omega_c})^{2n}}$$

$$= \frac{a^2}{1 + -\left[\frac{S^2}{\omega_c^2}\right]^n} = \begin{cases} \dfrac{a^2}{1 + \left[\frac{S^2}{\omega_c^2}\right]^n}, & \text{for } n \text{ even} \\ \dfrac{a^2}{1 - \left[\frac{S^2}{\omega_c^2}\right]^n}, & \text{for } n \text{ odd} \end{cases} \quad (10)$$

Setting the denominators equal to zero,

$$\frac{S}{\omega_c} = (\pm 1)^{1/2n} \quad (11)$$

Thus, the pole locations are the 2n roots of $\pm 1$, depending on whether the order is odd or even. These roots are located on a circle with radius $\omega_c$ centered at the origin of the S plane and have symmetry

with respect to both real and imaginary axes. For $n$ odd, a pair of roots are on the real axis and the rest are separated by $\pi/n$ radians. For $n$ even, a pair of roots are located $\pi/2n$ radians from the real axis and the rest are again separated by $\pi/n$ radians. No roots are on the imaginary axis for either even or odd $n$.

Let $p_1, \ldots, p_{2n}$ be the roots. From the symmetry of the pole locations, if $p_1, \ldots, p_n$ are the roots lying in the right-half plane, the left-half plane roots are $-p_1, \ldots, -p_n$. The magnitude-squared function can then be written as

$$H_n(S)H_n(-S) = \frac{a^2(-1)^n\omega_c^{2n}}{(S+p_1)\ldots(S+p_n)(S-p_1)\ldots(S-p_n)} \tag{12}$$

To be stable, $H_n(S)$ must have all its poles in the left-half plane, thus

$$H_n(S) = \frac{a\omega_c^n}{(S+p_1)\ldots(S+p_n)} . \tag{13}$$

The program is written with unity gain at DC, $(\omega=0)$, therefore $a = 1$.

In order to locate the poles as specified above, consider the following set of equations.

$$\begin{aligned} 1 &= -e^{\pm j\pi(2m-1)}, \quad m = 1, 2, \ldots, n; \text{ for } n \text{ even} \\ -1 &= -e^{\pm j2\pi k}, \quad k = 0, 1, \ldots, n; \text{ for } n \text{ odd} \end{aligned} \tag{14}$$

Substituting equations (14) into equation (11) yields

$$\begin{aligned} \left[\frac{S}{\omega_c}\right]_{\pm m} &= -e^{\pm j\pi(2m-1)/2n}, \quad m = 1, 2, \ldots, n; \text{ for } n \text{ even} \\ \left[\frac{S}{\omega_c}\right]_{\pm k} &= -e^{\pm j\pi k/n}, \quad k = 0, 1, \ldots, n; \text{ for } n \text{ odd} \end{aligned} \tag{15}$$

Equations (15) will give the pole locations as described above. Consider the form of equations (15)

$$S = -\omega_c e^{\pm j\theta} = \omega_c[-\cos\theta \pm j\sin\theta]. \tag{16}$$

From this relationship, it can be seen that the magnitude for each pole is $\omega_c$, regardless of the angle, and thus all the poles lie on a circle with radius $\omega_c$.

As an example consider a second order filter, n = 2.

$$\left[\frac{S}{\omega_c}\right]_{\pm m} = -e^{\pm j\pi(2m-1)/4} \qquad m = 1, 2$$

$$S_{\pm 1} = \omega_c \; \underline{/\pm 45^\circ}$$

$$S_{\pm 2} = \omega_c \; \underline{/\pm 135^\circ}$$

$$\theta = 45^\circ$$

The relationship of these roots about the circle of radius $\omega_c$ is illustrated in Figure 1. The angle $\theta$ is always measured from the negative real axis.

In the program, only the angle(s) less than 90° are considered so that poles lie in the left-half plane since poles in the left-half plane are stable. Putting $\theta = 45^\circ$ into equation (4) yields poles at $-0.707 \pm j0.707$. These locations are in the left-half plane.

In the program, only even order filters are considered.

Below are the values of $\theta$ for 1, 2, and 3 second order sections in cascade.

| Cascaded Sections N | Filter Order n | Angle $\theta$ |
|---|---|---|
| 1 | 2 | 45° |
| 2 | 4 | 22.5°, 67.5° |
| 3 | 6 | 75°, 45°, 15° |

These calculated angles are incorporated in the program in the order given above.

For $N$ second order sections there are $N$ $\theta$'s. Only one specific $\theta$ is used per stage, because each stage has only one set of pole locations.

The following is the procedure to derive the magnitude of the $i^{th}$ stage, where $i$ varies from 1 to $N$.

Given the normalized second order low-pass transfer function equation (4), we employ the low-pass to low-pass transformation for an arbitrary cutoff frequency $\omega_c$ given by

$$S \rightarrow \frac{s}{\omega_c} \tag{17}$$

For the $i^{th}$ stage, equation (4) becomes

$$H_i(S) = \frac{\omega_c^{\,2}}{S^2+2S\omega_c \cos\theta_i+\omega_c^{\,2}} \tag{18}$$

The extended bilinear Z transform, equation (2), is used to get to the digital domain. Employing equation (2) on equation (18) and substituting WDC for $\omega_c$ yields

$$H_i(Z) = \frac{WDC^2(Z^2+2Z+1)}{\frac{4}{T^2}(Z^2-2Z+1)+\frac{4}{T}(Z^2-1)WDC\cos\theta_i+WDC^2(Z^2+2Z+1)} \tag{19}$$

Putting the denominator of equation (19) in monic form yields the transfer function for the $i^{th}$ stage of the filter

$$H_i(Z) = \frac{K_{1i}(A_0Z^2+A_1Z+A_2)}{Z^2+B_{1i}Z+B_{2i}} \tag{20}$$

Equation (20) is the same as equation (1) with the exception of the subscripts. In equation (20)

$$A_0 = A_2 = 1$$

$$A_1 = 2$$

$$G_i = \frac{4}{T^2} + \frac{4}{T} WDC \cos\theta_i + WDC^2 \tag{21}$$

$$K_{1i} = \frac{WDC^2}{G_i}$$

$$B_{1i} = \frac{2WDC^2 - \frac{8}{T^2}}{G_i}$$

$$B_{2i} = \frac{\frac{4}{T^2} - \frac{4}{T} WDC\cos\theta_i + WDC^2}{G_i}$$

Letting $Z = e^{ST}$ and $S = j\omega$ and taking the magnitude of $H_i(j\omega)$ we have

$$|H_i(j\omega)| = K_{1i} \frac{\sqrt{(A_0\cos(2\omega T)+A_1\cos(\omega T)+A_2)^2+(A_0\sin(2\omega T)+A_1\sin(\omega T))^2}}{\sqrt{(\cos(2\omega T)+B_{1i}\cos(\omega T)+B_{2i})^2+(\sin(2\omega T)+B_{1i}\sin(\omega T))^2}} \tag{22}$$

This magnitude function is the same for both the Butterworth and the Chebychev filters where $i$ varies from 1 to N.

C. Chebychev Low-Pass Filter

The advantage of the Chebychev low-pass filter over the Butterworth low-pass filter is that the transition band of the response at frequencies greater than $\omega_c$ is sharper for the Chebychev low-pass filter. This is achieved by specifying a small percentage of ripple in the low-pass region. The amplitude of the ripple is specified by the quantity $\delta$ (labeled RIP in the program). Figures 6, 7 and 8 illustrate the rippling for second, fourth, and sixth order filters, respectively. The poles of the filter are found on an ellipse

described by two Butterworth circles of radii $A$ and $B$ with $A<B$. The location of the poles on the ellipse is a function of the ripple, $\delta$, and is given by the following equation:

$$B, A = \frac{1}{2}\left(\left(\sqrt{\varepsilon^{-2}+1}+\varepsilon^{-1}\right)^{1/2N} \pm \left(\sqrt{\varepsilon^{-2}+1}+\varepsilon^{-1}\right)^{-1/2N}\right) \tag{23}$$

where

$$\varepsilon = \left[\frac{1}{(1-\delta)^2} - 1\right]^{1/2} \tag{24}$$

$B$ is given for the plus sign and $A$ for the minus sign. The Chebychev ellipse then has major axis $B$ and minor axis $A$. The location of the $S$ plane poles on the ellipse is given by

$$\begin{aligned} \text{Real Part} &= A \cos\theta \\ \text{Imaginary Part} &= B \sin\theta \end{aligned} \tag{25}$$

The $\theta$'s are the same as given for the corresponding order Butterworth filter. An example of Chebychev pole locations is illustrated in Figure 2. For $A = 1/2$ and $B = 1$ in a fourth order filter, $\theta = 22.5°$ and $67.5°$. The Chebychev pole locations are determined from equations (23), (24), and (25).

The analog second order Chebychev low-pass filter is

$$H(S) = \frac{K_2 a}{S^2+K_8 S+K_2} \tag{26}$$

where

$$a = \left[\frac{1}{(1+\varepsilon^2)^{1/2}}\right]^{1/N} \tag{27}$$

$\varepsilon$ is calculated from equation (24) and $N$ is the number of second order sections. $K_8$ and $K_2$ are calculated by

$$K_8 = 2A\cos\theta \tag{28}$$

$$K_2 = A^2\cos^2\theta + B^2\sin^2\theta\ . \tag{29}$$

The substitution of the low-pass to low-pass transformation for some cutoff frequency $\omega_c$, equation (17), into equation (26) yields

$$H(S) = \frac{K_2 a\omega_c^{\ 2}}{S^2 + SK_8\omega_c + \omega_c^{\ 2}K_2} \tag{30}$$

Using the extended bilinear Z transform, equation (2), and substituting WDC for $\omega_c$ we have for any section

$$H(Z) = \frac{K_2 aWDC^2(Z^2+2Z+1)}{\frac{4}{T^2}(Z^2-2Z+1) + \frac{2}{T}K_8WDC(Z^2-1) + WDC^2K_2(Z^2+2Z+1)} \tag{31}$$

Collecting terms yields the following for the $i^{th}$ section

$$H_i(Z) = \frac{K_{1i}(A_0Z^2+A_1Z+A_2)}{Z^2+B_{1i}Z+B_{2i}} \tag{32}$$

where

$$A_0 = A_2 = 1$$

$$A_1 = 2$$

$$G_i = \frac{4}{T^2} + \frac{2}{T}WDC\cdot K_8 + WDC^2K_2$$

$$K_{1i} = \frac{aK_2WDC^2}{G_i} \tag{33}$$

$$B_{1i} = \frac{2WDC^2K_2 - \frac{8}{T^2}}{G_i}$$

$$B_{2i} = \frac{\frac{4}{T^2} - \frac{2}{T} WDC \cdot K_8 + WDC^2 K_2}{G_i}$$

i varies from 1 to N. These coefficients are used to find $|H_i(j\omega)|$ given in (22).

For applications where a sharper roll-off is required the Chebychev filters are used. The roll-off increases with n for any fixed $\varepsilon$. For fixed n, the roll-off decreases as $\varepsilon$ decreases. For small $\varepsilon$ the ripple width, $\delta$, is small, see equation (23), but so is the roll-off. For larger $\varepsilon$ the roll-off improves but the ripple width increases. In the first case the filter will be good at DC and low frequencies, unsatisfactory at high frequencies. The converse is true in the second case.

The above observations suggest the procedure to be used in selecting a Chebychev filter to match a set of specifications. The permissible ripple width specifies $\varepsilon$. With $\varepsilon$ fixed, select n to attain the required roll-off.

## II. Using the Program

The first data card read into the program contains the number of second order sections to be cascaded, N, and the type of filter desired, KN. N is equal to 1, 2, or 3, which corresponds to the $2^{nd}$, $4^{th}$, or $6^{th}$ order filter respectively. KN = 1 yields a Butterworth filter, while KN = 2 yields a Chebychev filter. The format on the N, KN card is 2I2. The second data card read in is the sampling interval T in F10.6 format. When choosing T, 1/T should be approximately equal to ten times the cutoff frequency, $\omega_c$. The third data card contains the value of $\omega_c$ in F10.4 format. For the Butterworth low-pass filter, $\omega_c$ is the -3db cutoff frequency. For the Chebychev filter the magnitude of the response is $1/(1+\epsilon^2)^{1/2} = 1 - \delta$ at $\omega = \omega_c$. $\omega$ is in radians. $\delta$ is the ripple factor.

If the desired filter is Chebychev, i.e., KN = 2, the next data card is the ripple factor (RIP) in F5.3 format. The filter response for all even order Chebychev low-pass filters passes through $1/(1+\epsilon^2)^{1/2} = 1 - \delta$ for $\omega = 0$ and $\omega_c$. For odd order filters, the magnitude is 1 for $\omega = 0$ and $1/(1+\epsilon^2)^{1/2} = 1 - \delta$ for $\omega = \omega_c$. This program produces only even order filters. If the desired filter is Butterworth, i.e., KN = 1, this data card is omitted from the data deck.

The final data card is the starting frequency (FREQ1) and the frequency increments (DELT) in radians. The format of the FREQ1, DELT card is 2F10.4. Determine DELT by the following:

$$\text{DELT} = \frac{\text{final frequency - starting frequency}}{1024}$$

This is necessary because there are 1024 frequency data points calculated in the program. Choose FREQ1 and DELT to insure that calculated values

will include the data of interest. For maximum efficiency of the program, DELT should be a multiple of $2^{-K}$ so no decimal to binary conversion errors are incurred.

The digital filter coefficients are computed and printed out for each second order section. The full filter magnitude response, as well as each section magnitude response, is printed for each of the frequency increments specified. When N = 1, the section magnitude response is the full filter magnitude response and is only printed once.

The program may be easily modified to incorporate a graphics display of the magnitude response. There is a comment card in the LPASS program indicating where the graphics subroutine call card should be inserted.

The program is written with input obtained via device 4 and output written to device 6. These numbers should be assigned to the appropriate devices prior to running the program.

The program was developed on the PDP-11/20 with a DOS/BATCH operating system. Trial runs frequently used a TTY terminal as well as a card reader for input (device 4); and a TTY terminal as well as a line printer for output (device 6).

Double precision arithmetic is employed. To decrease required memory storage, only the frequency interval values and the full magnitude response are saved. The section magnitude responses are printed out, but are not stored. The program will produce approximately 21 pages of output.

Shown below are sample deck set-ups for the Chebychev and Butterworth low-pass filters.

| Data Card | Format | Example |
|---|---|---|
| 1 | 2I2 | 0302 (3 sections Chebychev. low-pass) |
| 2 | F10.6 | 0.001 ($T = 0.001$) |
| 3 | F10.4 | 100 ($\omega_c = 100$ radians) |
| 4 | F5.3 | 0.10 (Ripple amplitude = 0.10) |
| 5 | 2F10.4 | 70 0.06 (Start at $\omega = 70$. Steps of 0.06 radians. Will finish just past $\omega = 131$ radians.) |
| 1 | 2I2 | 0201 (2 sections, 4[th] order, Butterworth) |
| 2 | F10.6 | 0.005 ($T = 0.005$) |
| 3 | F10.4 | 20 ($\omega_c = 20$ radians) |
| 4 | 2F10.4 | 0 0.04 (Start at $\omega = 0$, finish just past $\omega = 40$ radians in steps of 0.04 radians.) |

The following pages contain annotated examples of output data.

This is an example of the output for a $4^{th}$ ofder Butterworth low-pass filter with $T = 0.005$ and $\omega_c = 20$ radians. The starting frequency is 0 radians and the frequency increment is 0.04 radian.

WDC = 20.01668 WC = 20.00000 T = 0.50000E-02

| | | | | |
|---|---|---|---|---|
| FOR I = 1 | $A_0$ = 0.10000000E+01 | | $A_1$ = 0.20000000E+01 | |
| | $A_2$ = 0.10000000E+01 | | $K_1$ = 0.22869799E-02 | |
| | $B_1$ = 0.18219614E+01 | | $B_2$ = 0.83110937E+00 | |
| FOR I = 2 | $A_0$ = 0.10000000E+01 | | $A_1$ = 0.20000000E+01 | |
| | $A_2$ = 0.10000000E+01 | | $K_1$ = 0.24059972E-02 | |
| | $B_1$ = 0.19167786E+01 | | $B_2$ = 0.92640257E+00 | |

| W | H | H1 | H2 |
|---|---|---|---|
| 0.0000 | 0.10000E+01 | 0.10000E+01 | 0.10000E+01 |
| 0.0400 | 0.10000E+01 | 0.10000E+01 | 0.10000E+01 |
| 0.0800 | 0.99999E+00 | 0.99999E+00 | 0.10000E+01 |
| 0.1200 | 0.10000E+01 | 0.99997E+00 | 0.10000E+01 |
| 0.1600 | 0.10000E+01 | 0.99995E+00 | 0.10000E+01 |
| 0.2000 | 0.10000E+01 | 0.99993E+00 | 0.10000E+01 |
| 0.2400 | 0.10000E+01 | 0.99990E+00 | 0.10001E+01 |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |

I is the $i^{th}$ stage. I varies from 1 to N.

WDC is the prewarped cutoff frequency.

WC is the cutoff frequency.

T is the sampling interval.

$A_0$, $A_1$, and $A_2$ are the low-pass filter numerator coefficients.

$B_1$ and $B_2$ are the low-pass filter denominator coefficients.

$K_1$ is the gain factor.

W is the frequency.

H is the overall magnitude of the digital transfer function.

H1 is the magnitude of the digital transfer function ($1^{st}$ stage).

H2 is the magnitude of the digital transfer function ($2^{nd}$ stage).

H = H1*H2.

See Figure 4.

This is an example of the output for a $6^{th}$ order Chebychev low-pass filter (three second order stages cascaded) with T = 0.005 and $\omega_c$ = 20 radians. The starting frequency is 0 and the frequency increment is 0.04 radian. The ripple is equal to 0.100.

WDC = 20.01668 WC = 20.00000 T = 0.50000E-02

| | | | |
|---|---|---|---|
| A = 0.24783947 | B = 1.03025433 | $K_8$ = 0.12829114 | $K_2$ = 0.99443709 |
| A = 0.24783947 | B = 1.03025453 | $K_8$ = 0.35049793 | $K_2$ = 0.56142438 |
| A = 0.24783947 | B = 1.03025453 | $K_8$ = 0.47878908 | $K_2$ = 0.12841170 |

FOR I = 1 $A_0$ = 0.10000000E+01 $A_1$ = 0.20000000E+01
$A_1$ = 0.10000000E+01 $K_1$ = 0.23830688E-02
$B_1$ = -0.19774006E+01 $B_2$ = 0.98727357E+00

WDC2 = 0.40066761E+03 G(I) = 0.16142562E+06 A = 0.96548939E+00

FOR I = 2 $A_0$ = 0.10000000E+01 $A_1$ = 0.20000000E+01
$A_2$ = 0.10000000E+01 $K_1$ = 0.13321469E-02
$B_1$ = -0.19600541E+01 $B_2$ = 0.96557320E+00

WDC2 = 0.40066761E+03 G(I) = 0.16303127E+06 A = 0.96548939E+00

FOR I = 3 $A_0$ = 0.10000000E+01 $A_1$ = 0.20000000E+01

$A_2$ = 0.10000000E+01 $K_1$ = 0.30310788E-03

$B_1$ = -0.19519613E+01 $B_2$ = 0.95321709E+00

WDC2 = 0.40066761E+03 G(I) = 0.16388496E+06 A = 0.96548939E+00

| W | H | H1 | H2 | H3 |
|---|---|---|---|---|
| 0.0000 | 0.89998E+00 | 0.96549E+00 | 0.96549E+00 | 0.96547E+00 |
| 0.0400 | 0.89999E+00 | 0.96549E+00 | 0.96549E+00 | 0.96548E+00 |
| 0.0800 | 0.90001E+00 | 0.96550E+00 | 0.96551E+00 | 0.96547E+00 |
| 0.1200 | 0.90009E+00 | 0.96552E+00 | 0.96554E+00 | 0.96550E+00 |
| 0.1600 | 0.90016E+00 | 0.96555E+00 | 0.96558E+00 | 0.96551E+00 |
| 0.2000 | 0.90028E+00 | 0.96558E+00 | 0.96564E+00 | 0.96555E+00 |
| 0.2400 | 0.90042E+00 | 0.96563E+00 | 0.96571E+00 | 0.96559E+00 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |

WDC is the prewarped cutoff frequency.

WC is the cutoff frequency.

T is the sampling interval.

B, A $= \frac{1}{2}\left(\left(\sqrt{\varepsilon^{-2}+1}+\varepsilon^{-1}\right)^{1/2N} \pm \left(\sqrt{\varepsilon^{-2}+1}+\varepsilon^{-1}\right)^{-1/2N}\right)$

I is the $i^{th}$ stage, I varies from 1 to N.

$K_8 = 2A\cos\theta$.

$K_2 = A^2\cos^2\theta + B^2\sin^2\theta$.

$A_0$, $A_1$, and $A_2$ are the low-pass filter numerator coefficients.

$B_1$ and $B_2$ are the low-pass filter denominator coefficients.

$K_1$ is the gain factor.

WDC2 = $(\mathrm{WDC})^2$.

$G(I) = \frac{4}{T^2} + \frac{2}{T}\mathrm{WDC}\cdot K_8 + (\mathrm{WDC})^2 K_2$.

The A following G(I) is a $= \left[\frac{1}{1+\varepsilon^2}\right]^{1/2N}$

W is the frequency.

H is the overall magnitude of the digital transfer function.

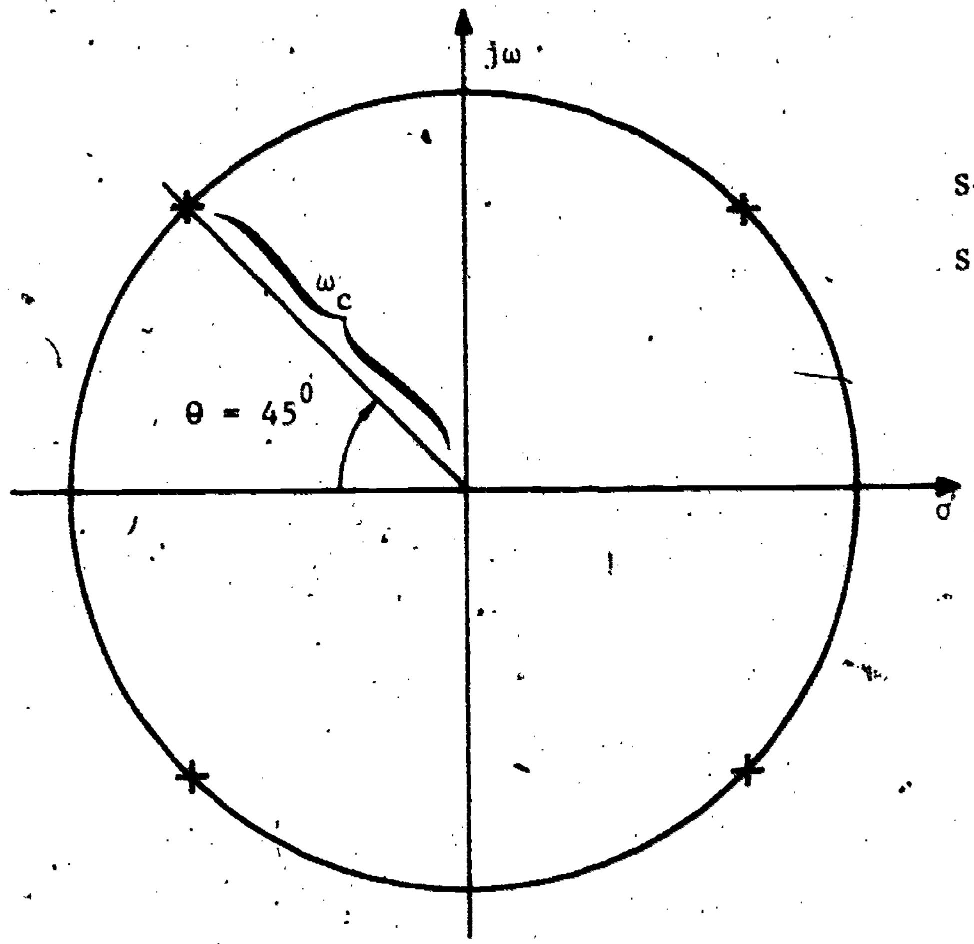
H1 is the magnitude of the digital transfer function (1st stage).

H2 is the magnitude of the digital transfer function (2nd stage).

H3 is the magnitude of the digital transfer function (3rd stage).

H = H1*H2*H3.

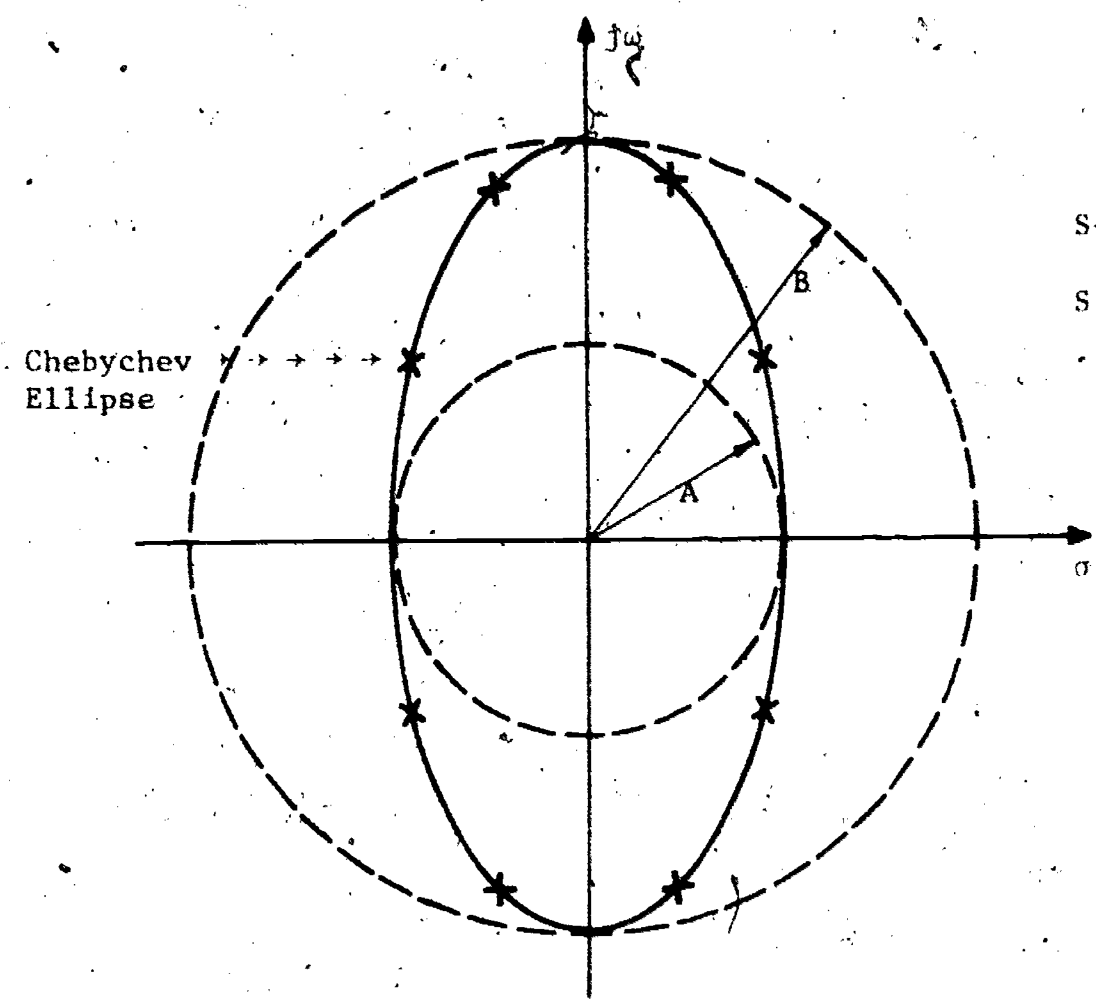
See Figure 8.




S-Plane

$S = \sigma + j\omega$

Figure 1




S-Plane

$S = \sigma + j\omega$

Figure 2

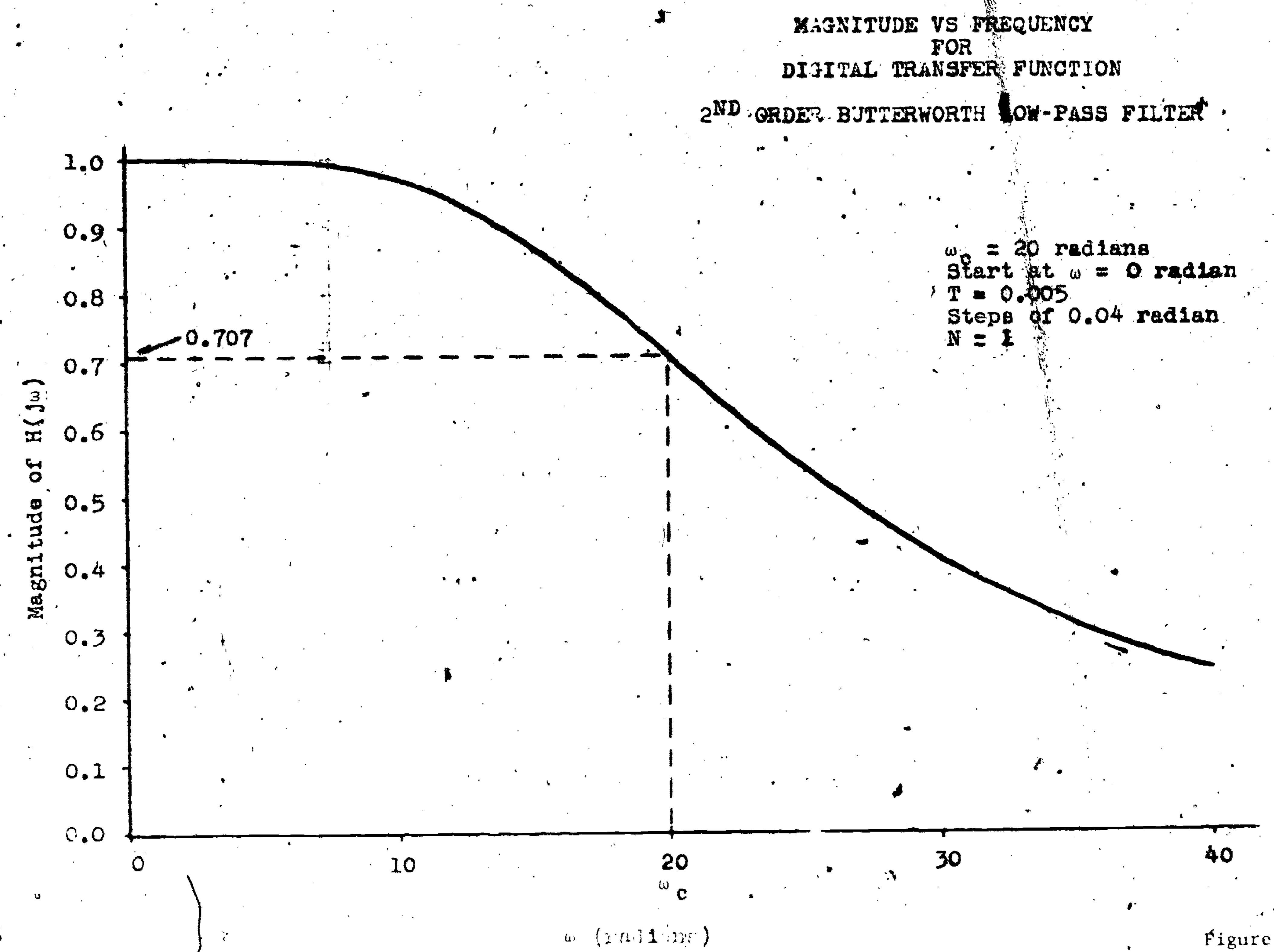
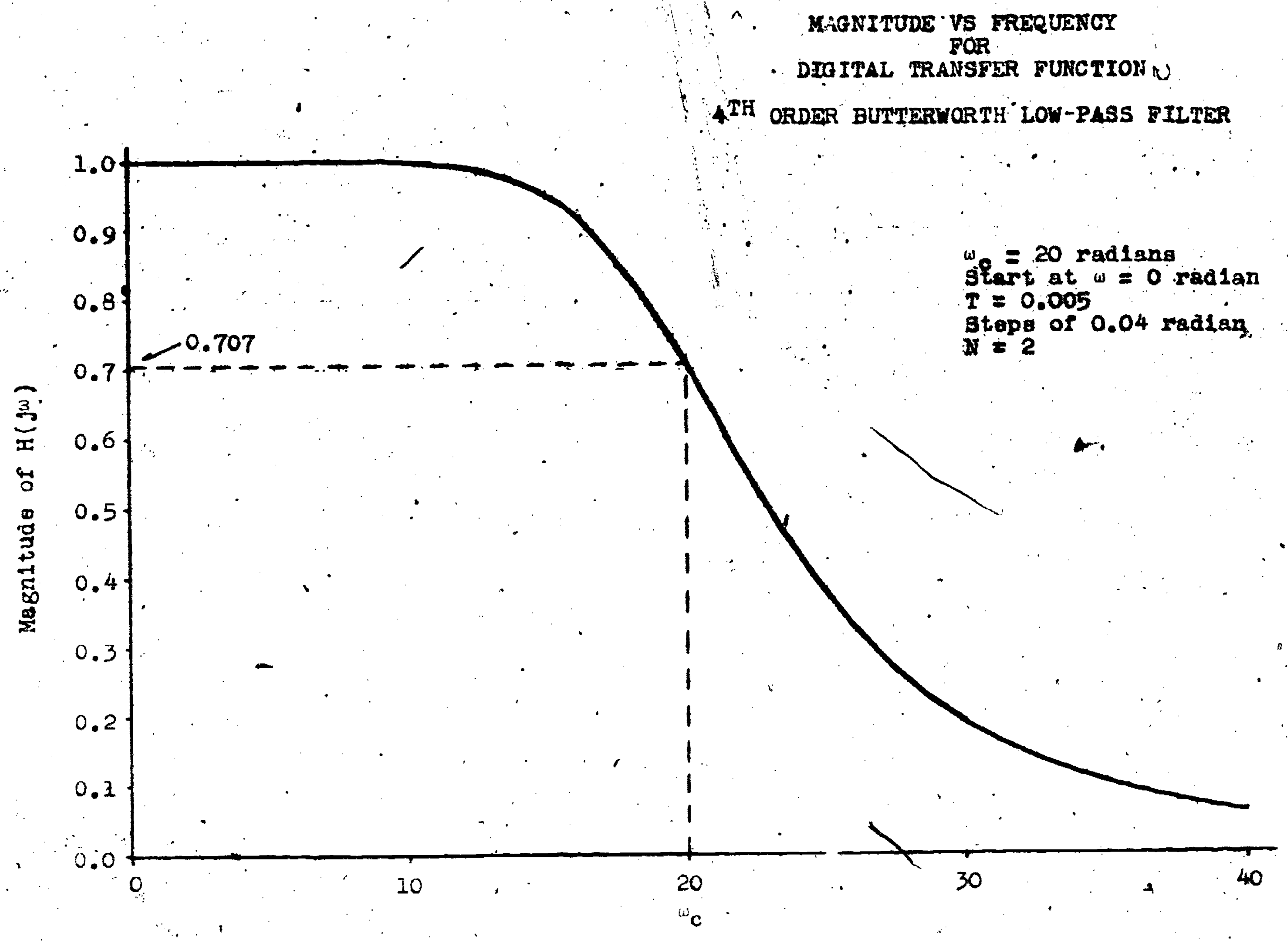

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION
2ND ORDER BUTTERWORTH LOW-PASS FILTER
$\omega_c$ = 20 radians
Start at $\omega$ = 0 radian
T = 0.005
Steps of 0.04 radian
N = 1
Magnitude of H(jω)
1.0
0.9
0.8
0.7
0.707
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
$\omega_c$
30
40
ω (radians)

Figure 3




ω (radians)

Figure 4

MA NITUDE VS FREQUENCY
FOR
DIG TAL TRANSFER FUNCTION

6TH ORDER BUTTERWORTH LOW-PASS FILTER.

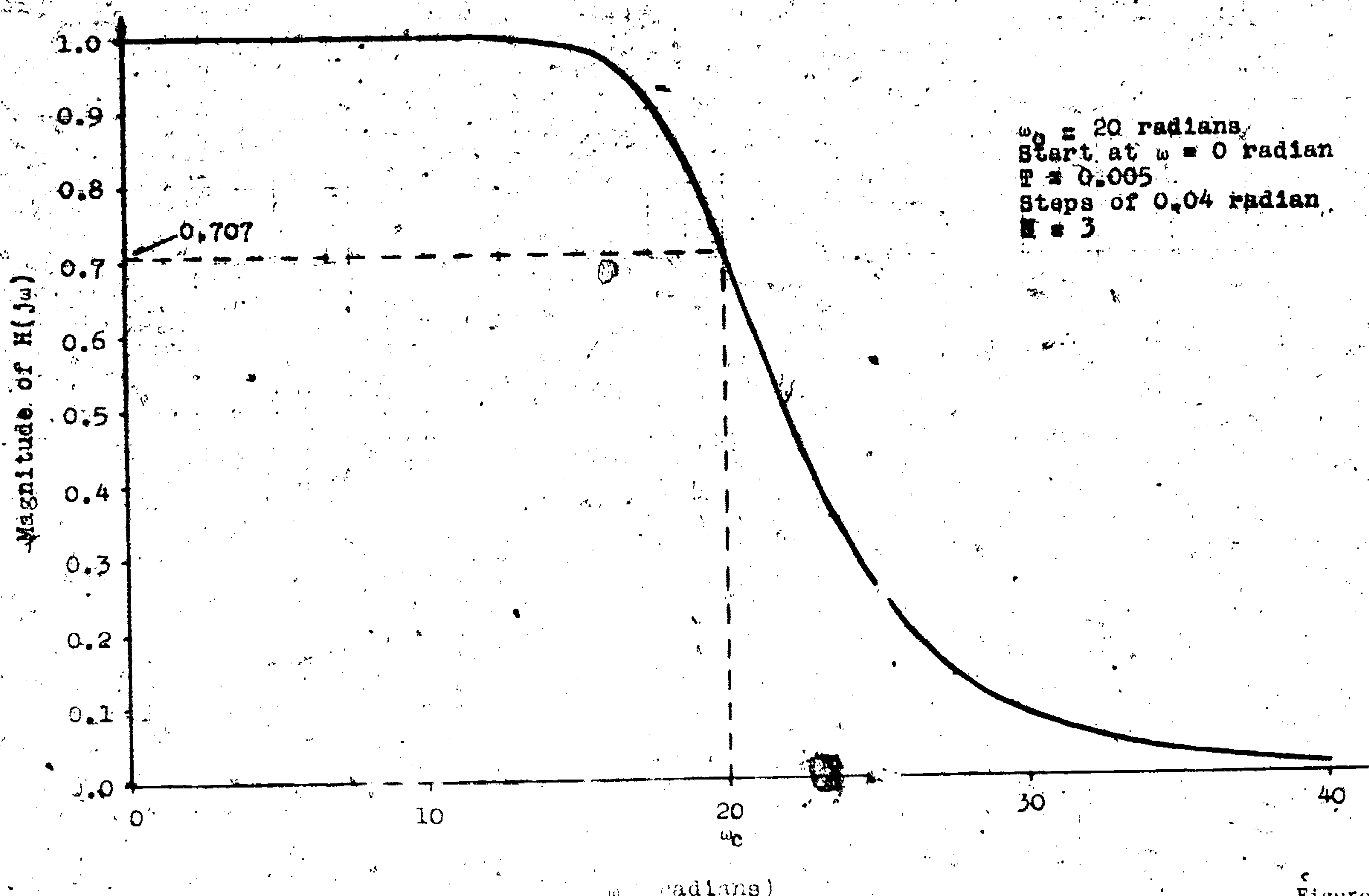
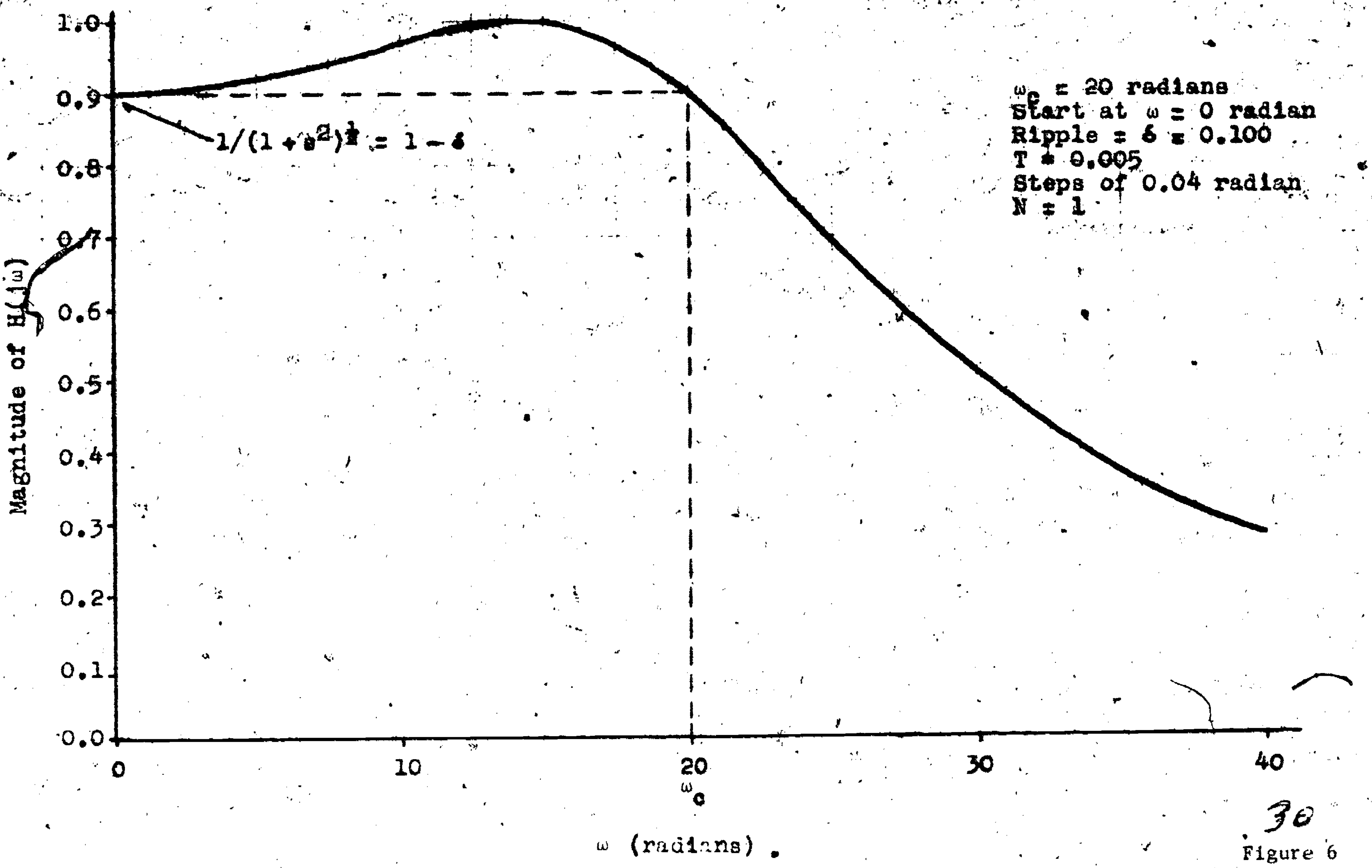



Figure 5

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

2ND ORDER CHEBYCHEV LOW-PASS FILTER




30

Figure 6

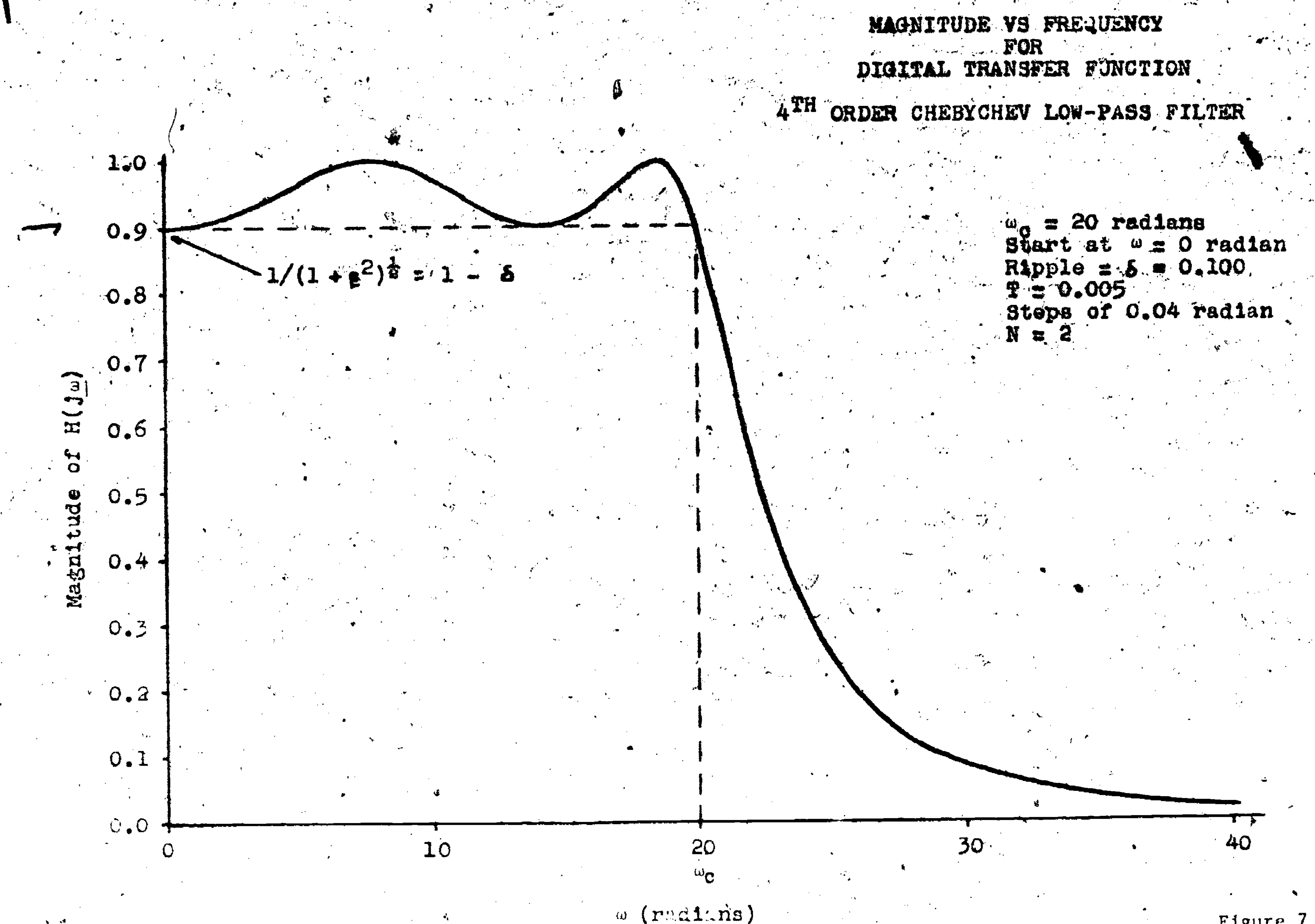
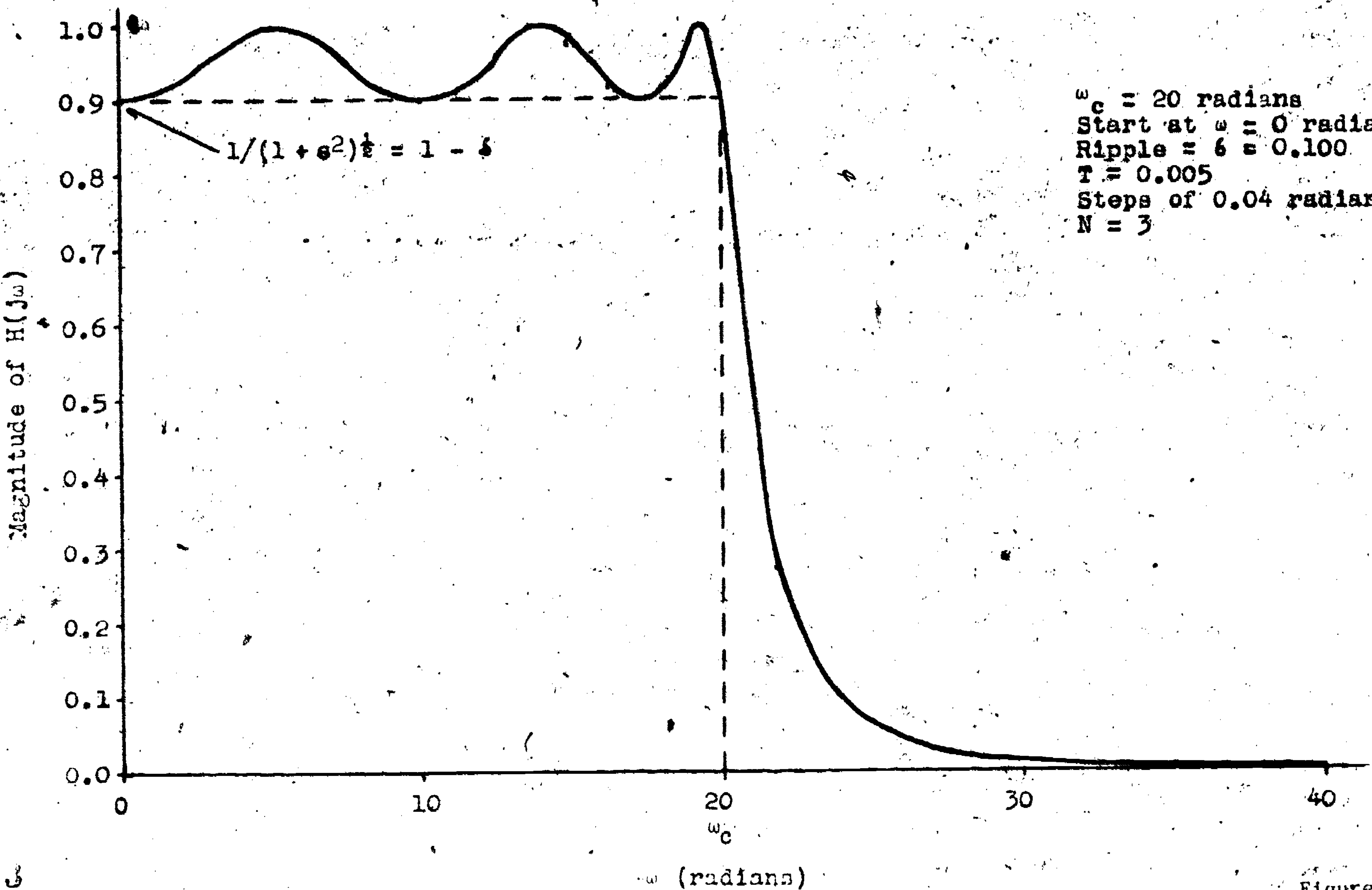

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION
4TH ORDER CHEBYCHEV LOW-PASS FILTER
$\omega_c$ = 20 radians
Start at $\omega$ = 0 radian
Ripple = $\delta$ = 0.100
T = 0.005
Steps of 0.04 radian
N = 2
$1/(1+\varepsilon^2)^{\frac{1}{2}} = 1 - \delta$
Magnitude of $H(j\omega)$
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
$\omega_c$
30
40
$\omega$ (radians)

Figure 7

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

$6^{TH}$ ORDER CHEBYCHEV LOW-PASS FILTER




$\omega_c$ = 20 radians
Start at $\omega$ = 0 radian
Ripple = $\delta$ = 0.100
T = 0.005
Steps of 0.04 radian
N = 3

Figure 8

## References

A. Budak, Passive and Active Network Analysis and Synthesis, Houghton Mifflin Co., Boston, 1974.

D. Childers and A. Durling, Digital Filtering and Signal Processing, West Publishing Company, New York, 1975.

J. J. D'Azzo and C. H. Houpis, Linear Control System Analysis and Design, McGraw-Hill, Inc., New York, 1975.

B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill, Inc., New York, 1969.

B. J. Leon and P. A. Wintz, Basic Linear Networks for Electrical and Electronics Engineers, Holt, Rinehart, and Winston, Inc., New York, 1970.

L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., New Jersey, 1975.

L. Weinberg, Network Analysis and Synthesis, McGraw-Hill, Inc., New York, 1962.

M. E. Van Valkenburg, Modern Network Synthesis, John Wiley & Sons, Inc., New York, 1960.

A FORTRAN IV DESIGN PROGRAM

FOR BUTTERWORTH AND

CHEBYCHEV BAND-PASS AND

BAND-STOP DIGITAL FILTERS

by

H. J. Markos

and

T. A. Brubaker

The authors are with the Electrical Engineering Department

at Colorado State University, Fort Collins, Colorado.

## INTRODUCTION

This report contains the documentation for the BPASS program. It consists of the design procedure used, a description of the program, and design examples using the program.

The purpose of the BPASS program is the design of either a maximally flat Butterworth or a Chébychev filter with equal ripple in the pass band. For each type of filter there is a choice of band-pass or band-stop filters. Starting with an analog filter, the bilinear Z transform is used to design an equivalent digital filter. The user enters the low-pass filter order, the type of filter desired, the sampling interval, the upper and lower cutoff frequencies, the starting frequency and frequency increment, and if a Chebychev filter is being designed, the ripple. The low-pass filter sections are transformed to second order band-pass or band-stop sections. Then the program generates the digital filter coefficients for up to six second order sections in cascade or up to a 12th order filter. The design is carried out in the frequency domain. The program calculates the transfer function coefficients for each second order section, the magnitude function for each section, and the final cascaded filter magnitude response over the frequency interval specified by the input.

The BPASS program, written in Fortran IV is supplied as a card deck with this report. The program is in the form of a subroutine and can be used as is by a call statement from the main program. Data may be input via cards with output available through a line printer. The input/output devices may be altered as explained in this report. Graphic routines may easily be appended to the program.

## I. Design Procedure

### A. Preliminary Discussion

One common method of designing a digital filter is to start with an analog transfer function $H(S)$ and transform it to the digital transfer function $H(Z)$.

The transfer function of a second order digital filter in the $Z$ domain is given by

$$H(Z) = \frac{K_1(A_0Z^2 + A_1Z + A_2)}{Z^2 + B_1Z + B_2} \tag{1}$$

where the A's and B's are the coefficients of the numerator and denominator respectively. This program will calculate the scale factor $K_1$ and the coefficients $A_0$, $A_1$, $A_2$, $B_1$, and $B_2$. The transformation used is the extended bilinear $Z$ transform

$$S \rightarrow \frac{2}{T}\left(\frac{Z - 1}{Z + 1}\right) \quad , \tag{2}$$

where $T$ is the sampling interval. When the extended bilinear $Z$ transform is employed, the desired frequencies must first be pre-warped to make them compatible with the digital filter. In the band-pass and band-stop filters, the upper and lower cutoff frequencies and the center frequency of the filter are of interest. Calling the upper and lower frequencies $\omega_u$, and $\omega_1$, respectively, the pre-warped upper (WDU), lower (WDL), and center (WDM) frequencies and the bandwidth between WDU and WDL are found by

$$WDU = \frac{2}{T}\tan\left(\frac{\omega_u T}{2}\right)$$

$$WDL = \frac{2}{T}\tan\left(\frac{\omega_l T}{2}\right)$$

$$WDM = \frac{2}{T}\tan\left[\frac{\sqrt{\omega_u\omega_l}\,T}{2}\right] \tag{3}$$

$$WB = WDU - WDL$$

$\omega_u$ and $\omega_l$ are specified by the designer and the prewarping is done by the program.

In the design procedure for all band-pass and band-stop filters of order $n'$, ($n'$ even), the program begins by first finding the poles for the corresponding $n'/2$ order low-pass filter. The low-pass filter is then transformed into a band-pass or band-stop filter of order $n'$, ($n' = 2n$).

B. Butterworth Band-Pass Filter

We start with a normalized second order low-pass Butterworth filter transfer function in the $S$ plane

$$H(S) = \frac{1}{S^2 + 2S\cos\theta + 1} \tag{4}$$

where the angle $\theta$ is in degrees (in the program) and may be found from the Butterworth circle and the relationship

$$S = e^{\pm j\pi(2m - 1)/2n} \tag{5}$$

where $n$ is the order of the low-pass filter and $m = 1, 2, \ldots, n$. This relationship is determined by the following procedure. By definition, a filter is nth order Butterworth low-pass if its gain characteristic is

$$|H_n(j\omega)|^2 = \frac{a^2}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}} \qquad (6)$$

where $a$ is the DC gain, $\omega_c$ is the desired cutoff frequency and $n$ is the order of the low-pass filter.

In the design, the poles of $H(S)$ must be found. The procedure is as follows:

$$|H_n(j\omega)|^2 = H_n(j\omega)\overline{H_n(j\omega)} = H_n(j\omega)H_n\overline{(j\omega)} = H_n(j\omega)H_n(-j\omega)$$

$$= [H(S)H(-S)]_{S=j\omega} = \left[\frac{a^2}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}\right]_{\omega = \frac{S}{j}} = \frac{a^2}{1 + \left(\frac{S}{j\omega_c}\right)^{2n}}$$

$$= \frac{a^2}{1 + \left[-\frac{S^2}{\omega_c^2}\right]^n} = \begin{cases} \dfrac{a^2}{1 + \left[\dfrac{S^2}{\omega_c^2}\right]^n}, & \text{for } n \text{ even} \\[2ex] \dfrac{a^2}{1 - \left[\dfrac{S^2}{\omega_c^2}\right]^n}, & \text{for } n \text{ odd} \end{cases} \qquad (7)$$

Setting the denominators equal to zero,

$$\frac{S}{\omega_c} = (\pm 1)^{1/2n} \qquad (8)$$

Thus, the pole locations are the $2n$ roots of $\pm 1$, depending on whether the low-pass filter order is odd or even. These roots are located on a circle with radius $\omega_c$ centered at the origin of the $S$ plane and have symmetry with respect to both real and imaginary axes.

For $n$ odd, a pair of roots are on the real axis and the rest are separated by $\pi/n$ radians. For $n$ even, a pair of roots are located $\pi/2n$ radians from the real axis and the rest are again separated by $\pi/n$ radians. No roots are on the imaginary axis, for either even or odd $n$.

Let $p_1,\ldots,p_{2n}$ be the roots. From the symmetry of the pole locations, if $p_1,\ldots,p_n$ are the roots lying in the right-half plane, the left-half plane roots are $-p_1,\ldots,-p_n$. The magnitude-squared function can then be written as

$$H_n(S)H_n(-S) = \frac{a^2(-1)^n \omega_c^{2n}}{(S + p_1)\ldots(S + p_n)(S - p_1)\ldots(S - p_n)} \qquad (9)$$

To be stable, $H_n(S)$ must have all its poles in the left-hand plane, thus

$$H_n(S) = \frac{a\omega_c^{\,n}}{(S + p_1)\ldots(S + p_n)} \qquad (10)$$

The program is written with unity gain at DC, ($\omega = 0$), therefore $a = 1$.

In order to locate the poles as specified above, consider the following set of equations.

$$\begin{aligned} 1 &= -e^{\pm j\pi(2m - 1)}, \quad m = 1, 2,\ldots, n;\ \text{for } n \text{ even} \\ -1 &= -e^{\pm j2\pi k}, \quad k = 0, 1,\ldots, n;\ \text{for } n \text{ odd} \end{aligned} \qquad (11)$$

Substituting equations (11) into equations (8) yields

$$\begin{aligned} \left[\frac{S}{\omega_c}\right]_{\pm m} &= -e^{\pm j\pi(2m - 1)/2n}, \quad m = 1, 2,\ldots, n;\ \text{for } n \text{ even} \\ \left[\frac{S}{\omega_c}\right]_{\pm k} &= -e^{\pm j\pi k/n}, \quad k = 0, 1,\ldots, n;\ \text{for } n \text{ odd} \end{aligned} \qquad (12)$$

Equations (12) will give the pole locations as described above. Consider the form of equations (12)

$$S = -\omega_c e^{\pm j\theta} = \omega_c[-\cos\theta \pm j\sin\theta] \qquad (13)$$

From this relationship, it can been seen that the magnitude for each pole is $\omega_c$, regardless of the angle, and thus all the poles lie on a circle with radius $\omega_c$.

As an example, consider a second order filter, n = 2.

$$[\frac{S}{\omega_c}]_{\pm m} = -e^{\pm j\pi(2m - 1)/4} \qquad m = 1, 2$$

$$S_{\pm 1} = \omega_c \underline{/\pm 45°}$$

$$S_{\pm 1} = \omega_c \underline{/\pm 135°}$$

$$\theta = 45°$$

The relationship of these roots about the circle of radius $\omega_c$ is illustrated in Figure 1. The angle $\theta$ is always measured from the negative real axis.

In the program, only the angle(s) less than 90° are considered so that the poles lie in the left-half plane because poles in the left-half plane are stable. Putting $\theta = 45°$ into equation (4) yields poles at -0.707 ±j0.707. These locations are in the left-half plane. From equations (12), for low-pass filter orders n = 1, 2,..., 6, the values of $\theta$ are given below.

| Low-Pass Filter Order | Angle | Second Order Cascaded Sections | Band-Pass Band-Stop Filter Order |
|---|---|---|---|
| n | θ | N | n' |
| 1 | 0° | 1 | 2 |
| 2 | 45° | 2 | 4 |
| 3 | 60°, 0° | 3 | 6 |
| 4 | 22.5°, 67.5° | 4 | 8 |
| 5 | 72°, 36°, 0° | 5 | 10 |
| 6 | 75°, 45°, 15° | 6 | 12 |

n is the order of the low-pass filter and is used to determine pole locations. n is also the number of second order band-pass or band-stop sections which results from the transformation of the low-pass filter sections and which will be cascaded to form the band-pass or band-stop filters of order n'. The transformation is explained below. The calculated angles are incorporated in the program in the order given above.

Given the normalized second order low-pass transfer function equation (4), we transform this low-pass into a band-pass transfer function for some bandwidth WB, and center frequency WDM by using the transform

$$S \rightarrow \frac{S^2 + WDM^2}{SWB}. \tag{14}$$

Equation (4) then transforms to a 4th order transfer function

$$H(S) = \frac{S^2 WB^2}{S^4 + S^3\, 2WB\cos\theta + S^2(2WDM^2 + WB^2) + S2WB\; WDM^2 \cos\theta + WDM^4}. \tag{15}$$

Using the root finding subroutine "POLRT" from the IBM Scientific Subroutine Package (SSP), the roots of the denominator of equation (15)

are found. (Note: POLRT has been attached to BPASS as a double precision subroutine and is included in the card deck). The roots found will be complex conjugate pairs. Calling the real and imaginary parts of the pairs $RE_1$, $AIM_1$, $RE_2$, $AIM_2$ equation (15) is factored to yield two cascaded second order sections

$$H(S) = \frac{SWB}{S^2 - 2SRE_1 + RE_1^2 + AIM_1^2} \cdot \frac{SWB}{S^2 - 2SRE_2 + RE_2^2 + AIM_2^2} \quad (16)$$

For each $\theta$ of a given N, the program calculates roots for both sections of equation (16) and labels them the ith and the ith + 1 section. If N, the number of second order sections specified, is even, the program will calculate N pairs of RE and AIM values or $2N = n'$ roots. If N is odd, the last value of $\theta$ is 0. Substituting $\theta = 0$ into equation (4) and factoring yields two identical first order sections, $1/(S + 1)$. The program will calculate N + 1 pairs of RE and AIM values, but because the last two pairs are the same due to the identical first order sections, the last pair will not be used.

Because both second order sections of equation (16) are of the same format, we will deal with only one section, the ith section and let

$$\begin{aligned} -2RE_i &= D_i \\ RE_i^2 + AIM_i^2 &= C_i \end{aligned} \quad (17)$$

The design of an n'th order band-pass or band-stop filter leads to n'/2 second order sections. Substituting equations (17) into one

section of equation (16) yields the transfer function for the ith section

$$H_i(S) = \frac{SWB}{S^2 + SD_i + C_i} . \tag{18}$$

The extended bilinear Z transform, equation (2), is used to get to the digital domain. Employing equation (2) on equation (18) yields $H_i(Z)$ for the ith second order section.

$$H_i(Z) = \frac{\frac{2}{T}WBZ^2 - \frac{2}{T}WB}{Z^2\left(\frac{4}{T^2} + \frac{2D_i}{T} + C_i\right) + Z\left(2C_i - \frac{8}{T^2}\right) + \left(\frac{4}{T^2} - \frac{2D_i}{T} + C_i\right)} \tag{19}$$

Putting the denominator of equation (19) in monic form yields the transfer function for the ith second order stage of the filter

$$H_i(Z) = \frac{K_{1i}(A_0Z^2 + A_1Z + A_2)}{Z^2 + B_{1i}Z + B_{2i}} \tag{20}$$

This equation is the same as equation (1) with the exception of the subscripts. For all four filter types discussed here, the scale factor, $K_1$, and coefficients $B_1$ and $B_2$ are a function of the section calculated, while the coefficients $A_0$, $A_1$, and $A_2$ are the same for all sections calculated. In going from equation (19) to equation (20) we have

$$A_0 = \frac{2}{T}WB$$

$$A_1 = 0$$

$$A_2 = -\frac{2}{T}WB$$

$$G_i = \frac{4}{T^2} + \frac{2D_i}{T} + C_i \qquad (21)$$

$$K_{1i} = \frac{1}{G_i}$$

$$B_{1i} = \frac{2C_i - \frac{8}{T^2}}{G_i}$$

$$B_{2i} = \frac{\frac{4}{T^2} - \frac{2D_i}{T} + C_i}{G_i}$$

Letting $Z = e^{ST} = e^{j\omega T}$ for $S = j\omega$ and taking the magnitude of $H_i(j\omega)$ we have

$$|H_i(j\omega)| = K_{1i}\frac{\sqrt{(A_0\cos(2\omega T) + A_1\cos(\omega T) + A_2)^2 + (A_0\sin(2\omega T) + A_1\sin(\omega T))^2}}{\sqrt{(\cos(2\omega T) + B_{1i}\cos(\omega T) + B_{2i})^2 + (\sin(2\omega T) + B_{1i}\sin(\omega T))^2}} \qquad (22)$$

The magnitude function, equation (22) is the same for all the filters discussed in this report.

C. Butterworth Band-Stop Filter

The design procedure is almost exactly the same as that of the Butterworth band-pass filter, except that the transformation to band-stop is the reciprocal of equation (14), i.e.

$$S \to \frac{SWB}{S^2 + WDM^2} \qquad (23)$$

and we find $H_i(S)$ to be

$$H_i(S) = \frac{S^2 + WDM^2}{S^2 + SD_i + C_i} \quad (24)$$

After employing the extended bilinear Z transform, equation (2), we have

$$A_0 = A_2 = \frac{4}{T^2} + WDM^2$$
$$A_1 = 2WDM^2 - \frac{8}{T^2} \quad (25)$$

and $B_{1i}$, $B_{2i}$, $K_{1i}$ are the same functions of $C_i$ and $D_i$ as in equation (21). These coefficients are then used in the calculation of equation (22) to find $|H_i(j\omega)|$.

D. Chebychev Band-Pass Filter

The Chebychev filter ripples with equal amplitude in the pass-band. The amount of ripple is specified by the quantity $\delta$ (labeled RIP in the program). The poles of the filter are found on an ellipse described by two Butterworth circles of radii A and B with $A < B$. The location of the poles on the ellipse is a function of the ripple and is given by the following equation:

$$B, A = \tfrac{1}{2}((\sqrt{\varepsilon^{-2} + 1} + \varepsilon^{-1})^{1/N} \pm (\sqrt{\varepsilon^{-2} + 1} + \varepsilon^{-1})^{-1/N}) \quad (26)$$

where

$$\varepsilon = \left[\frac{1}{(1 - \delta)^2} - 1\right]^{\frac{1}{2}} \quad (27)$$

and N is numerically equal to the order of the low-pass filter which is transformed to yield the band-pass filter. B is given

for the plus sign and A for the minus sign. The Chebychev ellipse then has major axis B and minor axis A. The location of the S plane poles on the ellipse is given by

$$\text{Real Part} = A\cos\theta$$

$$\text{Imaginary Part} = \beta\sin\theta \qquad (28)$$

The $\theta$'s are the same as given for the corresponding order Butterworth filter. An example of Chebychev pole locations is illustrated in Figure 2. For $A = \frac{1}{2}$ and $B = 1$ in a fourth order filter, $\theta = 22.5°$ and $67.5°$. The Chebychev pole locations are determined from equations (26), (27) and (28).

The analog second order Chebychev low-pass filter is

$$H(S) = \frac{K_2\left[\frac{1}{\sqrt{1 + \varepsilon^2}}\right]^{2/N}}{S^2 + K_8 S \quad K_2} \qquad (29)$$

$\varepsilon$ is calculated from equation (27) and N is equal to the order of the low-pass filter which is transformed to yield the band-pass filter. $K_8$ and $K_2$ are calculated by

$$K_8 = 2A\cos\theta \qquad (30)$$

$$K_2 = A^2\cos^2\theta + B^2\sin^2\theta \qquad (31)$$

The substitution of the low-pass to band-pass transformation, equation (14), into equation (29) yields

$$H(S) = \frac{S^2 WB^2 K_2\left[\frac{1}{\sqrt{1 + \varepsilon^2}}\right]^{2/N}}{S^4 + S^3 K_8 WB + S^2(2WDM^2 + K_2 WB^2) + SK_8 WDM^2 WB + WDM^4} \qquad (32)$$

After finding the roots of equation (32) and making the substitutions given by equations (17) we find the ith second order section

$$H_i(S) = \frac{SWK_3}{S^2 + SD_i + C_i},$$

$$K_3 = \sqrt{K_2}\left[\frac{1}{\sqrt{1+\varepsilon^2}}\right]^{1/N}. \tag{33}$$

Applying the extended bilinear Z transform equation (2) yields an equation of the form of equation (20) where

$$A_0 = 1$$

$$A_1 = 0$$

$$A_2 = -1$$

$$K_{1i} = \frac{2WB}{T} \cdot \frac{K_3}{G_i} \tag{34}$$

$B_{1i}$ and $B_{2i}$ are the same functions of $C_i$ and $D_i$ given by equations (21). These coefficients are then used in equation (22) to find $|H_i(j\omega)|$.

E. Chebychev Band-Stop Filter

Given equation (29) for $H(S)$ we apply the low-pass to band-stop transformation equation (23) to obtain the 4th order transfer function

$$H_i(S) = \frac{(S^2 + WDM^2)^2 K_2\left[\frac{1}{\sqrt{1+\varepsilon^2}}\right]^{2/N}}{K_2S^4 + S^3K_8WB + S^2(WB^2 + 2K_2WDM^2) + SK_8WDM^2WB + K_2WDM^4}. \tag{35}$$

$N$ is equal to the order of the low-pass filter which is transformed to yield the band-stop filter.

After finding the roots of equation (35) and making the substitutions given by equations (17) the ith second order section is

$$H_i(S) = \frac{(S^2 + WDM^2)K_3}{S^2 + SD_i + C_i},$$

$$K_3 = \sqrt{K_2}\left[\frac{1}{\sqrt{1 + \varepsilon^2}}\right]^{1/N} . \qquad (36)$$

Applying the extended bilinear Z transform equation (2) yields an equation of the form of equation (20) where

$$A_0 = A_2 = \frac{4}{T^2} + WDM^2$$

$$A_1 = 2WDM^2 - \frac{8}{T^2} \qquad (37)$$

$$K_{1i} = \frac{K_3}{G_i}$$

$B_{1i}$ and $B_{2i}$ are the same functions of $C_i$ and $D_i$ given by equations (21). These coefficients are then used in equation (22) to find $|H_i(j\omega)|$.

## II. Using the Program

The first data card read into the program contains the number of second order sections to be cascaded, N, and the type of filter desired, KN. N is equal to 1, 2,..., or 6, which corresponds to the order of the low-pass filter, and hence corresponds to the 2nd, 4th,..., or 12th order band pass or band stop filter respectively. KN is the type of filter desired. The values of KN specifies one of the four choices given by

| KN | Type |
|---|---|
| 1 | Butterworth Band-Pass |
| 2 | Butterworth Band-Stop |
| 3 | Chebychev Band-Pass |
| 4 | Chebychev Band-Stop |

The format on the N, KN card is 2I2.

The second data card read in is the sampling interval T in F10.6 format. When choosing T, 1/T should be approximately equal to ten times the center frequency (WDM).

The third data card read in contains the values of the upper and lower cutoff frequencies, $\omega_u$ and $\omega_1$, in 2F10.4 format. For the Butterworth filters, the cutoff frequencies are the -3db cutoff frequencies. For the Chebychev filters, the magnitude of the response is $1/(1 + \epsilon^2)^{\frac{1}{2}} = 1 - \delta$ at the cutoff frequencies. $\omega$ is in radians. $\delta$ is the ripple factor.

If the desired filter is Chebychev, i.e., KN = 3 or 4, the next data card contains the ripple (RIP) factor in F5.3 format. If the desired filter is Butterworth, i.e., KN = 1 or 2, this card is omitted from the data deck.

The final data card is the starting frequency (FREQ1) and the frequency increments (DELT) in radians. The format of the FREQ1, DELT card is 2F10.4. Determine DELT by the following:

$$DELT = \frac{\text{final frequency} - \text{starting frequency}}{1024} .$$

This is necessary because there are 1024 frequency data points calculated in the program. Choose FREQ1 and DELT to insure that calculated values will include the data of interest. For maximum efficiency of the program, DELT should be a multiple of $2^{-K}$ so no decimal to binary conversion errors are incurred.

The digital filter coefficients are computed and printed out for each second order section. The full filter magnitude response, as well as each section magnitude response, is printed for each of the specified frequency increments. When there is only one second order section, the section magnitude response is the full filter magnitude response and is only printed once.

The program may be easily modified to incorporate a graphics display of the magnitude response. There is a comment card in the BPASS program indicating where the graphics subroutine call card should be inserted.

The program is written with input obtained via device 4 and output written to device 6. These numbers should be assigned to the appropriate devices prior to running the program.

The program was developed on a PDP-11/20 with a DOS/BATCH operating system. Trial runs frequently used a TTY terminal as well as a card reader for input (device 4); and a TTY terminal as well as a line printer for output (device 6). Double precision arithmetic

is employed. To decrease required memory storage, only the frequency interval values and the full magnitude response are saved. The section magnitude responses are printed out, but are not stored. The program will produce approximately 21 pages of output.

Shown below are sample deck set-ups.

| Data Card | Format | Example |
|---|---|---|
| 1 | 2I2 | 0504 (5 sections Chebychev band-pass) |
| 2 | F10.6 | 0.002 ($T = 0.002$) |
| 3 | 2F10.4 | 60 40 ($\omega_u = 60$, $\omega_1 = 40$ radians) |
| 4 | F5.3 | 0.10 (Ripple amplitude = 0.10) |
| 5 | 2F10.4 | 0 0.1 (Start at $\omega = 0$. Steps of 0.1 radian. Will finish just past $\omega = 102$ radians). |
| 1 | 2I2 | 0401 (4 sections Butterworth band-pass) |
| 2 | F10.6 | 0.002 ($T = 0.002$) |
| 3 | 2F10.4 | 60 40 ($\omega_u = 60$, $\omega_1 = 40$ radians) |
| 4 | 2F10.4 | 0 0.1 (Start at $\omega = 0$. Steps of 0.1 radian. Will finish just past $\omega = 102$ radians). |

The following pages contain annotated examples of output data.

This is an example of the output for an 8th order Butterworth band-stop filter (0402) with $T = 0.002$, $\omega_u = 60$ radians, and $\omega_1 = 40$ radians. The starting frequency is 0 radian and the frequency increment is 0.1 radian.

WDU = 60.07210 WDL = 40.02135 WDM = 49.02902 WB = 20.05076
T = 0.20000E-02

THE ROOTS OF THE FILTER ARE GIVEN BELOW

REAL(1) = -8.52659476 IMAGINARY(1) = -44.46786740
REAL(2) = -9.99788916 IMAGINARY(2) = -52.140959[illegible]
REAL(3) = -4.55076557 IMAGINARY(3) = -59.01588870
REAL(4) = -3.12232691 IMAGINARY(4) = -40.49140500

THE COEFFICIENTS OF EACH DIGITAL FILTER SECOND ORDER SECTION ARE GIVEN BELOW

FOR I = 1 $A_0$ = 0.10024038E+07 $A_1$ = -0.19951923E+07
$A_2$ = 0.10024038E+07 $K_1$ = 0.98125481E-06
$B_1$ = -0.19584863E+01 $B_2$ = 0.96653295E+00

FOR I = 2 $A_0$ = 0.10024038E+07 $A_1$ = -0.19951923E+07
$A_2$ = 0.10024038E+07 $K_1$ = 0.97769448E-06
$B_1$ = -0.19498774E+01 $B_2$ = 0.96090048E+00

FOR I = 3 $A_0$ = 0.10024038E+07 $A_1$ = -0.19951923E+07
$A_2$ = 0.10024038E+07 $K_1$ = 0.98755180E-06
$B_1$ = -0.19681836E+01 $B_2$ = 0.98202353E+00

FOR I = 4 $A_0$ = 0.10024038E+07 $A_1$ = -0.19951923E+07
$A_2$ = 0.10024038E+07 $K_1$ = 0.99216788E-06
$B_1$ = 0.1[illegible]E+01 $B_2$ = 0.9[illegible]E+00

| W | H | H1 | H2 | H3 | H4 |
|---|---|---|---|---|---|
| 0.0000 | 0.10000E+01 | 0.11726E+01 | 0.85284E+00 | 0.68611E+00 | 0.14575E+01 |
| 0.1000 | 0.10000E+01 | 0.11726E+01 | 0.85284E+00 | 0.68611E+00 | 0.14575E+01 |
| 0.2000 | 0.10000E+01 | 0.11726E+01 | 0.85284E+00 | 0.68611E+00 | 0.14575E+01 |
| 0.3000 | 0.99999E+00 | 0.11726E+01 | 0.85283E+00 | 0.68610E+00 | 0.14575E+01 |
| 0.4000 | 0.10000E+01 | 0.11726E+01 | 0.85283E+00 | 0.68609E+00 | 0.14576E+01 |
| 0.5000 | 0.10000E+01 | 0.11726E+01 | 0.85282E+00 | 0.68609E+00 | 0.14576E+01 |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| . | . | . | . | . | . |

WDU is the prewarped upper frequency.

WDL is the prewarped lower frequency.

WDM is the prewarped center frequency.

WB is the bandwidth, WDU - WDL.

T is the sampling interval.

The Real and Imaginary part of the roots of the filter are given next.

I is the ith stage. I varies from 1 to N.

$A_0$, $A_1$, $A_2$ are the Butterworth band-stop filter numerator coefficients.

$K_1$ is the gain factor.

$B_1$, and $B_2$ are the Butterworth band-stop filter denominator coefficients.

W is the frequency

H is the overall magnitude of the digital transfer function

H1 is the magnitude of the digital transfer function (1st stage).

H2 is the magnitude of the digital transfer function (2nd stage).

H3 is the magnitude of the digital transfer function (3rd stage).

H4 is the magnitude of the digital transfer function (4th stage).

See Figure 4.

This is an example of the output for an 8th order Chebychev band-pass filter (0403) with T = 0.002, $\omega_u$ = 60 radians, and $\omega_1$ = 40 radians. The starting frequency is 0 radian, the frequency increment is 0.1 radian, and the ripple is 0.1.

WDU = 60.07210 WDL = 40.02135 WDM = 49.02902 WB = 20.05076
T = 0.20000E-02

A = 0.37642105 B = 1.06850027 $K_8$ = 0.69553541 $K_2$ = 0.28813942
A = 0.37642105 B = 1.06850027 $K_8$ = 0.28810020 $K_2$ = 0.99524620

THE ROOTS OF THE FILTER ARE GIVEN BELOW
REAL(1) = -3.19528085 IMAGINARY(1) = -44.97792290
REAL(2) = -3.77772492 IMAGINARY(2) = -53.17662810
REAL(3) = -1.15829660 IMAGINARY(3) = -40.10115290
REAL(4) = -1.73001696 IMAGINARY(4) = -59.89456990

THE COEFFICIENTS OF EACH DIGITAL FILTER SECOND ORDER SECTION ARE GIVEN BELOW

FOR I = 1 $A_0$ = 0.10000000E+01 $A_1$ = 0.00000000E+00
$A_2$ = -0.10000000E+01 $K_1$ = 0.10395603E-01
$B_1$ = -0.19792607E+01 $B_2$ = 0.98732565E+00

FOR I = 2 $A_0$ = 0.10000000E+01 $A_1$ = 0.00000000E+00
$A_2$ = -0.10000000E+01 $K_1$ = 0.10375296E-01
$B_1$ = -0.19737935E+01 $B_2$ = 0.98504460E+00

FOR I = 3 $A_0$ = 0.10000000E+01 $A_1$ = 0.00000000E+00
$A_2$ = -0.10000000E+01 $K_1$ = 0.19406846E-01
$B_1$ = -0.19889723E+01 $B_2$ = 0.99538493E+00

FOR I = 4 $A_0$ = 0.10000000E+01 $A_1$ = 0.00000000E+00
$A_2$ = -0.10000000E+01 $K_1$ = 0.19346636E-01
$B_1$ = -0.19788675E+01 $B_2$ = 0.99312838E+00

| W | H | H1 | H2 | H3 | H4 |
|---|---|---|---|---|---|
| 0.0000 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 |
| 0.1000 | 0.12493E-12 | 0.51560E-03 | 0.36886E-03 | 0.12106E-02 | 0.54265E-03 |
| 0.2000 | 0.19990E-11 | 0.10312E-02 | 0.73773E-03 | 0.24211E-02 | 0.10853E-02 |
| 0.3000 | 0.10121E-10 | 0.15468E-02 | 0.11066E-02 | 0.36318E-02 | 0.16280E-02 |
| 0.4000 | 0.31991E-10 | 0.20625E-02 | 0.14755E-02 | 0.48426E-02 | 0.21707E-02 |
| 0.5000 | 0.78116E-10 | 0.25783E-02 | 0.18445E-02 | 0.60536E-02 | 0.27134E-02 |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| . | . | . | . | . | . |

WDU is the prewarped upper frequency.

WDL is the prewarped lower frequency.

WDM is the prewarped center frequency.

WB is the bandwidth, WDU - WDL.

T is the sampling interval.

$B, A = \frac{1}{2}((\sqrt{\varepsilon^{-2} + 1} + \varepsilon^{-1})^{1/N} \pm (\sqrt{\varepsilon^{-2} + 1} + \varepsilon^{-1})^{-1/N})$

$K_8 = 2A\cos(\theta)$

$K_2 = A^2\cos^2(\theta) + B^2\sin^2(\theta)$

The Real and Imaginary part of the roots of the filter are given next.

I is the Ith stage. I varies from 1 to N.

$A_0$, $A_1$, $A_2$ are the Chebychev band-pass filter numerator coefficients.

$K_1$ is the gain factor.

$B_1$, and $B_2$ are the Chebychev band-pass filter denominator coefficients.

W is the frequency.

H is the overall magnitude of the digital transfer function.

H1 is the magnitude of the digital transfer function (1st stage).

H2 is the magnitude of the digital transfer function (2nd stage).

H3 is the magnitude of the digital transfer function (3rd stage).

H4 is the magnitude of the digital transfer function (4th stage).

See Figure 5.

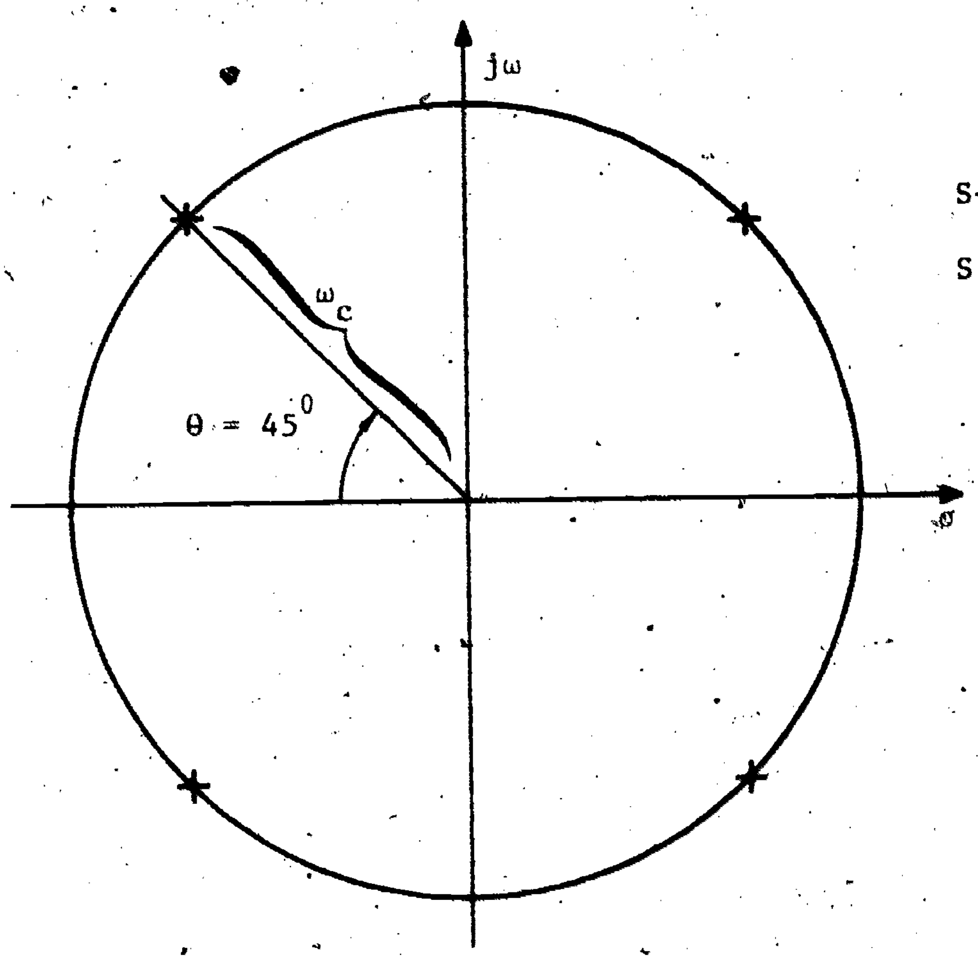



Figure 1

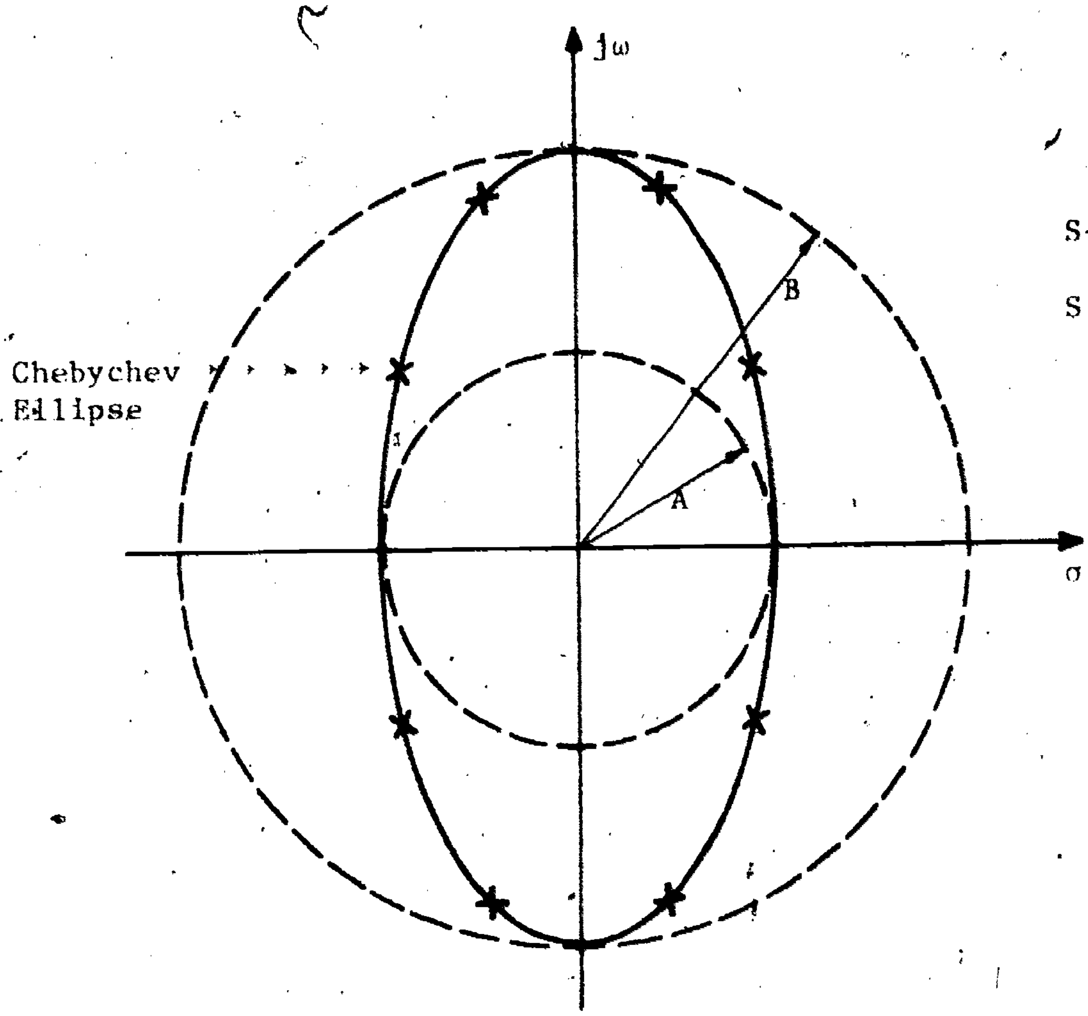



Figure 2

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

8th ORDER BUTTERWORTH BAND-PASS FILTER

N = 4
Start at $\omega$ = 0 radian
T = 0.002
Steps of 0.1 radian
$\omega_u$ = 60 radians
$\omega_l$ = 40 radians

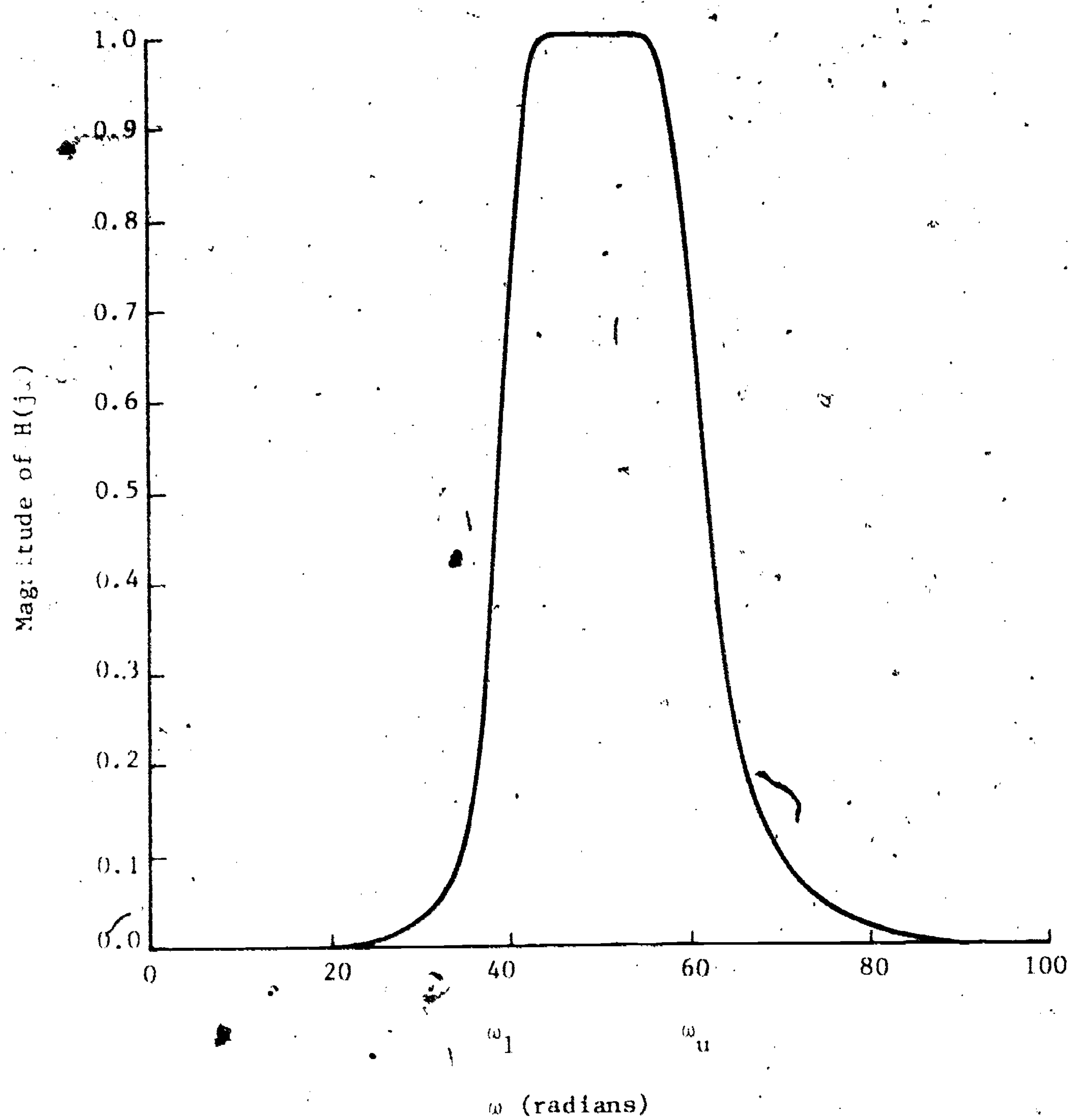



Figure 3

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

$8^{th}$ ORDER BUTTERWORTH BAND-STOP FILTER

N = 4
Start at $\omega$ = 0 radian
T = 0.002
Steps of 0.1 radian
$\omega_u$ = 60 radians
$\omega_l$ = 40 radians

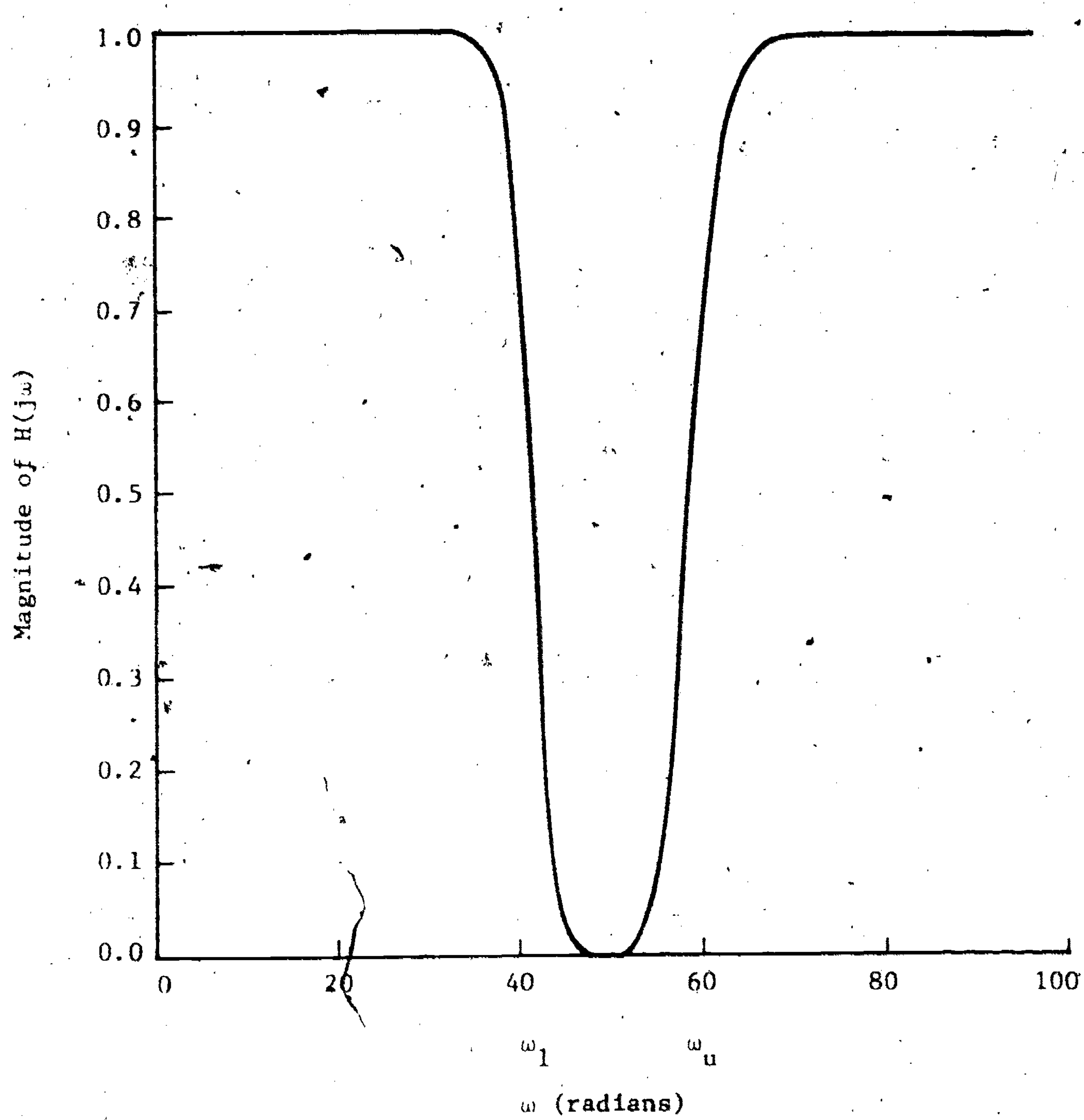



Figure 4

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

$8^{th}$ ORDER CHEBYCHEV BAND-PASS FILTER

$N = 4$
Start at $\omega = 0$ radian
Ripple = $\sigma = 0.100$
$T = 0.002$
Steps of 0.1 radian
$\omega_u = 60$ radians
$\omega_1 = 40$ radians

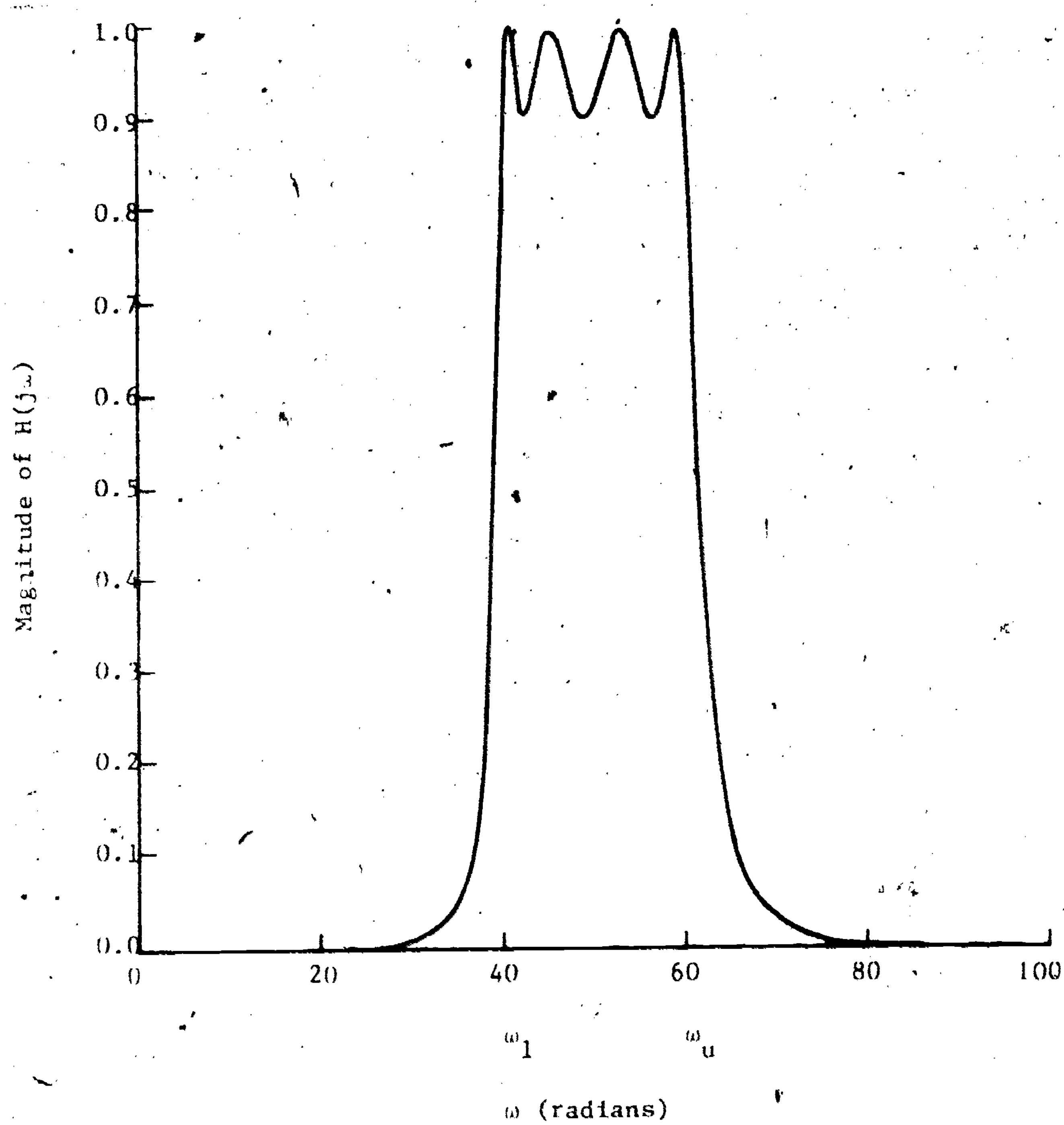



Figure 5

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

$10^{th}$ ORDER CHEBYCHEV BAND-STOP FILTER

N = 5
Start at $\omega$ = 0 radian
Ripple = $\sigma$ = 0.100
T = 0.002
Steps of 0.1 radian
$\omega_u$ = 60 radians
$\omega_l$ = 40 radians

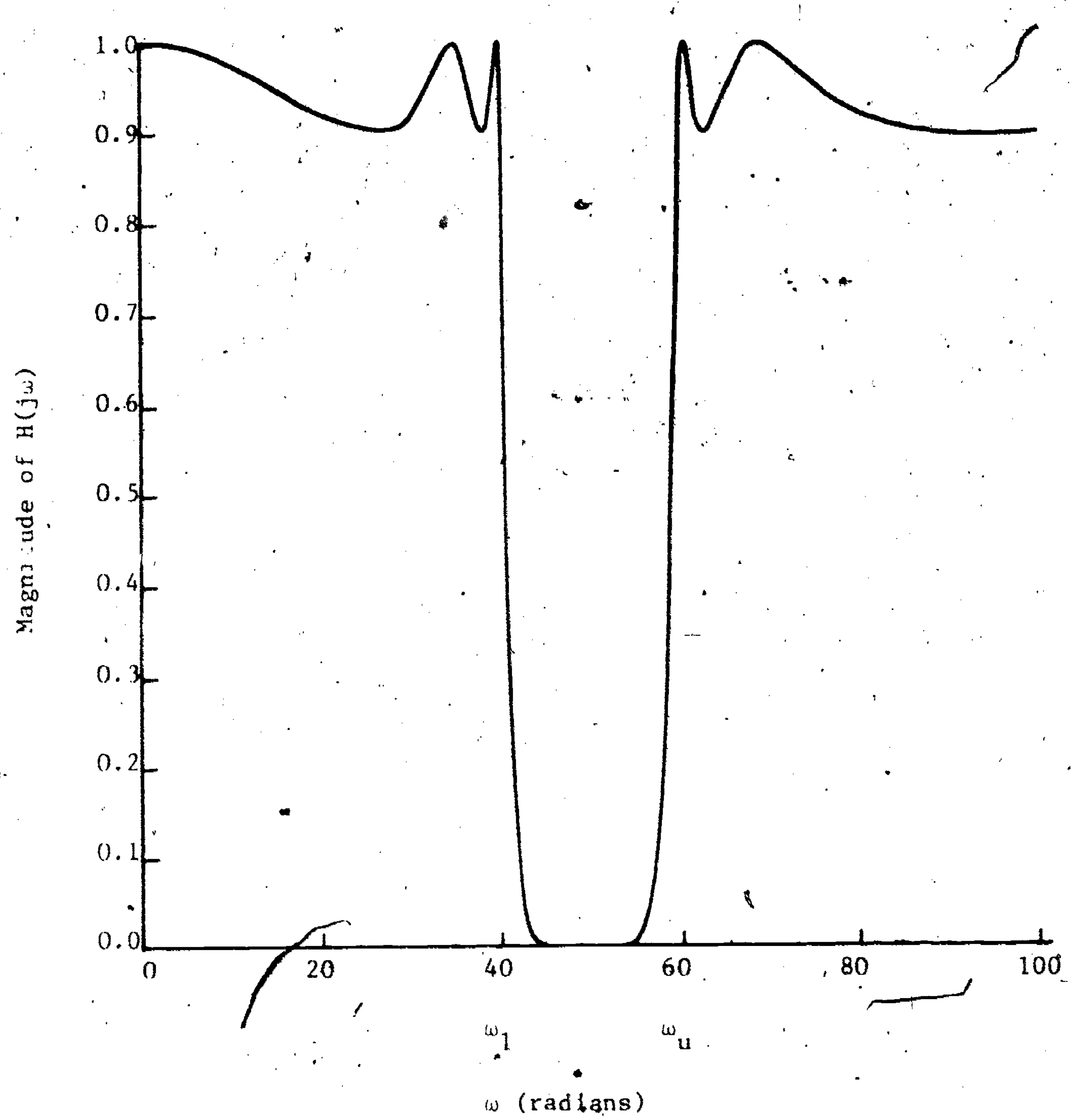



Figure 6

References

A. Budak, Passive and Active Network Analysis and Synthesis, Houghton Mifflin Co., Boston, 1974.

D. Childers and A. Durling, Digital Filtering and Signal Processing, West Publishing Company, New York, 1975,

J. J. D'Azzo and C. H. Houpis, Linear Control System Analysis and Design, McGraw-Hill, Inc., New York, 1975.

B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill, Inc., New York, 1969.

B. J. Leon and P. A. Wintz, Basic Linear Networks for Electrical and Electronics Engineers, Holt, Rinehart, and Winston, Inc., New York, 1970.

L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., New Jersey, 1975.

M. E. Van Valkenburg, Modern Network Synthesis, John Wiley & Sons, New York, 1960.

L. Weinberg, Network Analysis and Synthesis, McGraw-Hill, Inc., New York, 1962.

IBM S/360, Scientific Subroutine Package Version 3, Publication GH 20-0205-4, pp. 181,182.

# Programs for Weighted Least Squares Design of Nonrecursive and Recursive Digital Filters

by

F. L. Mann

E. Reuss

and

Thomas A. Brubaker

Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80523

# INTRODUCTION

This report describes how to use two programs for the weighted least squares design of nonrecursive and recursive digital filters. First the theoretical aspects are considered and the design equations are developed. The signal model in this work is assumed to be a polynomial because the state model is simple. However, the theory is easily extended to include any signal model represented by a linear differential equation.

Then the operation of the two programs is described along with examples to illustrate their operation.

LEAST SQUARES DESIGN OF NONRECURSIVE DIGITAL FILTERS

The design of nonrecursive and recursive digital filters using weighted least squares is based on a model for the input signal. The most common models that are currently in use are differential equations. The subsequent representation of the differential equation by a first-order vector differential equation leads to the concept of a state variable and the state space representation for a system. By use of the proper formulation, a continuous signal represented by a differential equation can be described by a first-order vector difference equation or a discrete time state space model. References that describe the essential aspects of state variables are by De Russo, Roy and Close [1] and Chen [2].

Since signals from laboratory instruments are not usually described by a differential equation, approximations often utilize a polynomial. For this reason, the vector form of a polynomial approximation will be used in this report. The reader should be aware that this can be generalized to include any signal model that can be represented as a linear time-varying differential equation. Later in the paper a scalar model representing a Gaussian time signal will also be utilized to develop a time-varying filter that can be used for reducing the base-line error and for the initial separation of signal components.

To develop the polynomial model let the signal $z(t)$ be represented by a polynomial of order m. At the time $t = nT$ the state vector for the signal is given by

$$\underline{z}(nT) = \begin{bmatrix} z \\ \dot{z} \\ \cdot \\ \cdot \\ D^m z \end{bmatrix}_{t=nT} \tag{1}$$

Redefining the state vector as

$$\underline{x}(nT) = \begin{bmatrix} z \\ T\dot{z} \\ \frac{T^2}{2!}\ddot{z} \\ \cdot \\ \cdot \\ \frac{T^m}{m!} D^m z \end{bmatrix}_{t=nT} \tag{2}$$

the use of a Taylor series representation for each element of x(nT) now permits the state of system at t=(n+h)T to be described in terms of the state at t=nT by the relationship

$$\underline{x}[(n + h)T] = \Phi[h]x[nT] \tag{3}$$

where $\Phi[h]$ is the mxm state transition matrix with elements

$$\Phi_{ij}[h] = \frac{j!}{i!(j-i)!} h^{j-i} \quad \begin{array}{l} 0\leq i\leq m \\ i\leq j\leq m \end{array} \tag{4}$$

$$= 0 \qquad i<j$$

The state transition matrix $\Phi[h]$ satisfies all of the relationships for general state transition matrices with the important two being for this work

$$\Phi[-h] = [\Phi(h)]^{-1} \tag{5}$$

and

$$\Phi[m]\ \Phi[p] = \Phi[m + p] \tag{6}$$

At this point, these models can be utilized in the design process.

## Design of Nonrecursive Filters

Let the input signal start at time t=0 and assume that the signal over a finite data window is approximated by a polynomial z(t). Defining the state of the signal by (2), then the state of the signal at time t=(n+h)T is given in terms of the signal at time t=nT by (3). Given that the signal starts at t=0, the first $\ell$ observations are each defined as

$$\underline{m}[jT] = M\underline{x}[jT] + \underline{v}[jT] \quad j=0,1,2,\ldots \ell-1 \tag{7}$$

where M is a row matrix that relates the measurable state variables to the actual measurements. The elements of each noise vector $\underline{v}[jT]$ are the measurement noises. In general these are taken as random variables with zero mean and the time dependent autocovariance matrix

$$R[jT] = E[\underline{v}(jT)\underline{v}^t(jT)] \tag{8}$$

However, in most laboratory systems only the data is available and the derivatives are not measurable. Furthermore, the noise covariance matrix is usually not known and for scalar measurements the noise variance is assumed constant and the noise samples uncorrelated. For the remainder of this paper, only scalar measurements and uncorrelated measurement noise with time-invariant statistics will be assumed. The mean value of the noise will be taken as zero and the variance as $\sigma^2$. The results are easily extended to vector measurements and to measurement noise with time varying statistics.

For $\ell$ observations, the total observation vector at $t=nT$ is defined as

$$\underline{m}_t[nT] = \begin{bmatrix} m[nT] \\ m[(n-1)T] \\ \cdot \\ \cdot \\ M[(n-\ell+1)T] \end{bmatrix} \tag{9}$$

This vector now forms a data window of $\ell$ data points. For an estimate of the data at $t=nT$ the use of the expression

$$\underline{x}[(n-j)T] = \Phi(-j)\ \underline{x}[nT], \tag{10}$$

is combined with (9) to yield

$$\underline{m}_t[nT] = \begin{bmatrix} M \\ M\Phi(-1) \\ \cdot \\ \cdot \\ M\Phi[-\ell+1] \end{bmatrix} \underline{x}[nT] + \underline{v}_t[nT] \tag{11}$$

The matrix of constants H[nT] is now defined as

$$H[nT] = \begin{bmatrix} M \\ M\Phi(-1) \\ \cdot \\ \cdot \\ M\Phi[-\ell+1] \end{bmatrix} \tag{12}$$

so that (11) can be written as

$$\underline{m}_t[nT] = H[nT]\ \underline{x}[nT] + \underline{v}_t[nT] \tag{13}$$

The elements in H[nT] are constants given by

$$H_{ij} = (-i)^j \quad \begin{matrix} 0 \le i < \ell \\ 0 \le j < q \end{matrix} \tag{14}$$

where $\ell$ is the size of the data window and q is the order of the polynomial plus one. When i=j=0 the value of (14) is one. Note that since (12) is a matrix of constants there is no need to make the matrix a function of time. However, in the derivation for the recursive filters this provides a method for separating different H matrices.

The optimal estimate of the data at t=nT is now given in terms of weighted least squares or minimum variance because this form is utilized in the derivation of the recursive filters. If the covariance matrix for the total observation vector is

$$R_t(nT) = E[\underline{v}_t(nT)\ \underline{v}_t^t(nT)] \tag{15}$$

the optimal minimum variance estimate is

$$\hat{\underline{x}}[nT] = \hat{W}[nT]\ m_t[nT] \tag{16}$$

where $\hat{W}[nT]$ is a series of constant weights given by

$$\hat{W}[nT] = \left[ H^t(nT)[R_t(nT)]^{-1}H(nT) \right]^{-1} H^t(nT)[R_t(nT)]^{-1} \tag{17}$$

For uncorrelated noise with constant variance $\sigma^2$ (15) is a diagonal matrix with elements $\sigma^2$ and (17) reduces to

$$\hat{W}[nT] = [H^t(nt)\ H(nt)]^{-1}\ H^t(nT) \tag{18}$$

This is the same result obtained using conventional least squares when the noise covariance is a diagonal matrix of equal constants. The reader should be aware that the estimate vector is an estimate of the data and all of the derivatives in the model. However, all of the weights in the derivative terms must be scaled by the scale factors in the state vector defined by (2).

The covariance matrix for the estimate given by (16) is given by

$$\hat{S}(nT) = \hat{W}(nT)\, R_t(nT)\, \hat{W}^t(nT) \tag{19}$$

Substituting (17) into (19) now yields

$$\hat{S}(nT) = [H^t(nT)\,[R_t(nT)]^{-1}\, H(nT)]^{-1} \tag{20}$$

which simplifies further to

$$\hat{S}(nT) = [H^t(nT)\, H(nT)]^{-1}\, \sigma^2 \tag{21}$$

for the uncorrelated noise.

By defining a delay or prediction factor $\alpha$, estimates of the data $\alpha T$ units behind or ahead of the point $t=nT$ can be done. To do this the total observation vector is written as

$$\underline{m}_t(nT) = H_\alpha(nT)\, \underline{x}[(n-\alpha)T] + \underline{v}_t(nT) \tag{22}$$

where $H_\alpha(nT)$ now takes the form

$$H_\alpha(nT) = \begin{bmatrix} M\Phi(-\alpha) \\ M\Phi(-\alpha-1) \\ \cdot \\ \cdot \\ M\Phi(-\ell-\alpha+1) \end{bmatrix} \tag{23}$$

The individual elements of the matrix now become

$$H_\alpha(nT)_{ij} = (\alpha-i)^j \qquad \begin{matrix} 0\le i\le \ell \\ 0<j<q \end{matrix} \tag{24}$$

The form for the optimal estimate now utilizes $H_\alpha(nT)$ in (18). The covariance of the estimate uses $H_\alpha(nT)$ in (21).

## Example

For a five-point window with $\alpha=0$, the optimal weight matrix and the covariance matrix for the estimate are shown in Table 1 for a third-order polynomial fit. For $\alpha=2$, the estimate of the data in the middle of the window is obtained and weight matrix and covariance matrices are shown in Table 2. This case corresponds to weights given in reference [3].

In practice, the design of nonrecursive filters is initiated by specifying the size of the window which is the number of rows in the H matrix, the order of the polynomial approximation which is one less than the number of columns in the H matrix and $\alpha$ which determines the coefficient values in the H matrix. This makes the design suitable for use with interactive graphics since only three parameters need be specified to generate the weight and covariance matrices.

If the model of the signal process is modified by additive uncorrelated driving noise, the variance terms of the driving noise fade the memory so that past data has less effect on the estimate. This can be thought of as uncertainty in the signal model. The concept is particularly important in recursive filter design. For non-recursive filters, it can also be utilized and can be useful when using a non-recursive filter to initialize a recursive filter.

If the model includes driving noise, the state at time $t=nT$ is given by

$$\underline{x}[nT] = \Phi[-1]\ \underline{x}[(n-1)T] + \underline{w}[(n-1)T] \tag{25}$$

where $\underline{w}[(n-1)T]$ is a sample from a noise process with mean zero and covariance matrix Q. This noise process is assumed to be white. In

terms of the total measurement vector, $\underline{m}_t[nT]$ now becomes

$$\underline{m}_t[nT] = H[nT]\ \underline{x}[nT] + \underline{p}_t(nT) + \underline{v}_t[nT] \qquad (26)$$

where $\underline{p}_t[nT]$ is the total noise vector due to the driving noise. For scalar measurements this is given by

$$\underline{p}_t[nT] = \begin{bmatrix} 0 \\ -M\Phi[-1]\ \underline{w}[(n-1)T] \\ -M\Phi[-2]\ \underline{w}[(n-1)T] - M\Phi[-1]\ \underline{w}[(n-2)T] \\ \vdots \end{bmatrix} \qquad (27)$$

The total noise vector that corrupts the total measurement vector is now defined as

$$\underline{r}_t[nT] = \underline{p}_t[nT] + \underline{v}_t[nT] \qquad (28)$$

For scalar measurements the covariance matrix for $\underline{r}_t[nT]$ has diagonal elements whose value increases down the diagonal. Since the diagonal elements are not the same, the minimum variance expressions of (16) and (17) must be employed to find optimal linear estimates.

<u>Example</u>

To illustrate how the driving noise affects the filter weights, consider a zero order process which is equivalent to estimating a signal of constant value. For a three-point filter with scalar measurements, the total noise vector is

$$\underline{r}_t[nT] = \begin{bmatrix} v[nT] \\ v[(n-1)T] - \underline{w}[(n-1)T] \\ v[(n-2)T] - \underline{w}[(n-2)T] - \underline{w}[(n-1)T] \end{bmatrix} \qquad (29)$$

If the variance of the measurement noise is $\sigma^2$ and of the driving noise $\sigma_1^2$ the covariance matrix of $\underline{r}_t[nT]$ is

$$R_t[nT] = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 + \sigma_1^2 & 0 \\ 0 & 0 & \sigma^2 + 2\sigma_1^2 \end{bmatrix} \tag{30}$$

The resulting weight matrix obtained using minimum variance is

$$W[nT] = [a_0 \; a_1 \; a_2] \tag{31}$$

where

$$a_0 = \frac{(\sigma_1^2 + \sigma^2)\;(2\sigma_1^2 + \sigma^2)}{D}, \tag{32}$$

$$a_1 = \frac{\sigma^2(2\sigma_1^2 + \sigma^2)}{D}, \tag{33}$$

and

$$a_2 = \frac{\sigma^2(\sigma_1^2 + \sigma^2)}{D}, \tag{34}$$

where D is given by

$$D = (2\sigma_1^2 + \sigma^2)\;(\sigma_1^2 + \sigma^2) + \sigma^2\;(2\sigma_1^2 + \sigma^2) + \sigma^2(\sigma_1^2 + \sigma^2) \tag{35}$$

The terms in the weight matrix satisfy the following inequality

$$a_0 \geq a_1 \geq a_2 \tag{36}$$

For $\sigma_1^2 = 0$, the equalities hold and $a_0 = a_1 = a_2 = 1/3$ which are the well known weights to estimate a mean. If $\sigma_1^2$ is not zero, the inequalities hold and as $\sigma_1^2$ becomes large with respect to $\sigma^2$, $a_0$ approaches one and $a_1$ and $a_2$ become smaller so that fading is

introduced. The covariance of the estimate is a scalar given by

$$\hat{S}(nT) = \frac{(\sigma_1^2 + \sigma^2)\ (2\sigma_1^2 + \sigma^2)\ \sigma^2}{D} = C\sigma^2 \qquad (37)$$

where C is a dimensionless constant whose value approaches one as $\sigma_1^2$ becomes larger than $\sigma^2$. Thus for $\sigma_1^2 \gg \sigma^2$ only the current observation is effectively used and the variance of the estimate is approximately $\sigma^2$.

The fading obviously reduces the signal-to-noise enhancement of a nonrecursive digital filter. On the other hand, fading can be used to reduce the deterministic error due to a nonexact model. In practice, fading in nonrecursive filters is not often utilized. Instead, the window length is more typically used as a design parameter. In future work, however, it may be desirable to further explore the relationship between fading and the size of the data window to achieve improved designs when nonexact signal models are employed.

## LEAST SQUARES DESIGN OF RECURSIVE DIGITAL FILTERS

The fixed memory or nonrecursive filter design is now extended to recursive digital filters that utilize all of the data. The result is a recursive form that is usually called the Kalman filter. While there are a variety of derivations for the Kalman filter, starting with a fixed memory filter using a polynomial model with driving noise gives the result in such a way to give the reader greater intuition about how the filter works.

The derivation of the recursive filter is started by using a signal model

$$\underline{x}[nT] = \Phi[1]\ \underline{x}[(n-1)T] + \underline{w}[(n-1)T] \qquad (38)$$

and the scalar observation or measurement model

$$y[nT] = M\underline{x}[nT] + \underline{v}[nT]. \tag{39}$$

In (38) and (39) the terms $w[(n-1)T]$ and $v[nT]$ are the driving noise and the measurement noise. These noise terms have zero mean and are uncorrelated with themselves and each other. Given the n+1 measurements starting at t=0, the total observation vector at time t=nT is given in terms of $\underline{x}[nT]$ as

$$\underline{y}[nT] = H[nT] + \underline{p}_t[nT] + \underline{v}_t[nT] \tag{40}$$

In (40) the matrix H[nT] is

$$H[nT] = \begin{bmatrix} M \\ \hline M\Phi[-1] \\ \cdot \\ \cdot \\ \cdot \\ M\Phi[-n] \end{bmatrix} \tag{41}$$

the vector $\underline{p}_t[nT]$ is

$$\underline{p}_t[nT] = \begin{bmatrix} 0 \\ -M\Phi[-1]\ w[(n-1)T] \\ -M\sum_{j=1}^{2} \Phi[-(3-j)]\ \underline{w}[(n-j)T] \\ \cdot \\ \cdot \\ -M\sum_{j=1}^{n} \Phi[-(n+1-j)]\ \underline{w}[(n-j)T] \end{bmatrix} \tag{42}$$

and $\underline{v}_t[nT]$ is the total measurement noise vector. Defining the sum

$$\underline{r}_t[nT] = \underline{p}_t[nT] + \underline{v}_t[nT] \tag{43}$$

as the total noise vector with autocovariance

$$R_t[nT] = E\{\underline{r}_t[nT]\ \underline{r}_t^t[nT]\} \tag{44}$$

the optimal estimate using linear minimum variance is given by

$$\hat{\underline{x}}[nT] = W[nT]\ \underline{y}_t[nT] \tag{45}$$

where W[nT] is the weight matrix

$$W[nT] = \left[H^t[nT]\left[R_t[nT]\right]^{-1} H[nT]\right]^{-1} H^t[nT]\left[R_t[nT]\right]^{-1} \tag{46}$$

The covariance of the estimate is

$$\hat{S}[nT] = W[nT]\ R_t[nT]\ W^t[nT] \tag{47}$$

Substitution of (46) and (47) gives an alternate form

$$\hat{S}[nT] = \left[H^t[nT]\left[R_t[nT]\right]^{-1} H[nT]\right]^{-1} \tag{48}$$

which allows the alternate form for (46) to be written as

$$W[nT] = \hat{S}[nT]\ H^t[nT]\ R_t\left[[nT]\right]^{-1} \tag{49}$$

The forms given by (46), (47), (48) and (49) apply to the remaining estimates used in the derivation and the reader should remember them.

Next the prediction or forecast of the signal state X[(n+1)T] is found by first writing the total observation vector $\underline{y}_t[nT]$ as

$$Y_t[nT] = H_1[nT]\ \underline{x}[(n+1)T] + \underline{p}_{1t}[nT] + \underline{v}_t[nT] \tag{50}$$

where $H_1[nT]$ is given as

$$H_1[nT] = H[nT]\ \Phi[-1] \tag{51}$$

and $\underline{p}_{1t}[nT]$ is

n+1 x n+1 matrix with elements $\sigma^2$. Taking the covariance of $\underline{p}_{1t}[nT]$ and performing some algebraic manipulation gives

$$E\{\underline{p}_{1t}[nT]\ \underline{p}_{1t}^{t}[nT]\} = E\{\underline{p}_1[nT]\ \underline{p}_1^{t}[nT]\} + H_1[nT]\ QH_1^{t}[nT] \tag{57}$$

where Q is the diagonal covariance matrix of the driving noise vector $\underline{w}[nT]$. This matrix is usually assumed to be time invariant. Thus, from (57)

$$R_{1t}[nT] = R_t[nT] + H_1[nT]\ QH_1^{t}[nT] \tag{58}$$

Substituting (58) into (56) now gives

$$S_1[(n+1)T] = W_1[nT]\ R_t[nT]\ W_1^{t}[nT] + W_1[nT]\ H_1[nT]\ QH_1^{t}[nT]\ W_1^{t}[nT] \tag{59}$$

If $X_1[(n+1)T]$ is an unbiased estimate then the constraint relationship

$$W_1[nT]\ H_1[nT] = I \tag{60}$$

must be satisfied [4] so that (59) can be simplified to

$$S_1[(n+1)T] = W_1[nT]\ R_t[nT]\ W_1^{t}[nT] + Q \tag{61}$$

Also recognizing that $\underline{x}_1[(n+1)T]$ can be written as

$$\underline{x}_1[(n+1)T] = \Phi[1]\ \hat{\underline{x}}[nT] \tag{62}$$

the weight matrix $W_1[nT]$ is

$$W_1[nT] = \Phi[1]\ W(nT). \tag{63}$$

Substitution at (58) into (61) and applying (47) now gives the final form for $S_1[(n+1)T]$ as

$$S_1[(n+1)T] = \Phi[1]\ \hat{S}[nT]\ \Phi^{t}[nT] + Q \tag{64}$$

The recursion is now formulated. When the observation at $t = (n+1)T$ arrives the new total observation vector in terms of the signal state

$\underline{x}[(n+1)T]$ is

$$\underline{y}_t[(n+1)T] = H[(n+1)T] + \underline{p}_t[(n+1)T] + \underline{v}_t[(n+1)T]. \tag{65}$$

The estimate is given by

$$\hat{\underline{x}}[(n+1)T] = \hat{S}[(n+1)T]\, H^t[(n+1)T]\, R_t[(n+1)T]^{-1} \tag{66}$$

with the covariance

$$\hat{S}[(n+1)T] = \left[H^t[(n+1)T] \left[R_t[(n+1)T]\right]^{-1} H[(n+1)T]\right]^{-1} \tag{67}$$

where $R_t[(n+1)T]$ is the covariance matrix of the total measurement noise vector

$$\underline{r}_t[(n+1)T] = \underline{p}_t[(n+1)T] + \underline{v}_t[(n+1)T] \tag{68}$$

First the recursion for the covariance matrix is found. Given that

$$\underline{p}_t[(n+1)T] = \begin{bmatrix} -M\Phi[-1]\ \underline{w}[nT] \\ -M \sum_{j=1}^{2} \Phi[-(3-j)]\ \underline{w}[(n+1-j)T] \\ \vdots \\ -M \sum_{j=1}^{n+1} \Phi[-(n+1-j)]\ \underline{w}[(n+1-j)T] \end{bmatrix} \tag{69}$$

substitution of (52) into (69) gives

$$\underline{p}_t[(n+1)T] = \begin{bmatrix} 0 \\ \hline \underline{p}_{1t}[nT] \end{bmatrix} \tag{70}$$

Next $H[(n+1)T]$ is given as

$$H[(n+1)T] = \begin{bmatrix} M \\ \hline M\Phi[-1] \\ \hline \vdots \\ M\Phi[-(n+1)] \end{bmatrix} \tag{71}$$

which can be rewritten with the aid of (41) as

$$H[nT] = \begin{bmatrix} M\Phi[1] \\ - \\ H[nT] \end{bmatrix} \Phi[-1] \tag{72}$$

The covariance matrix for $\underline{v}_t[(n+1)T]$ is now an n+2 x n+2 diagonal matrix with elements $\sigma^2$ so that $R_t[(n+1)T]$ is seen to be

$$R_t[(n+1)T] = \left[\begin{array}{c|c} \sigma^2 & 0 \\ \hline 0 & R_{1t}[nT] \end{array}\right] \tag{73}$$

Since this is a diagonal matrix its inverse is

$$R_t[(n+1)T] = \left[\begin{array}{c|c} \frac{1}{\sigma^2} & 0 \\ \hline 0 & \left[R_{1t}[nT]\right]^{-1} \end{array}\right] \tag{74}$$

Substitution of (74) and (72) into (67) and performing the matrix multiplication now yields

$$\hat{S}[(n+1)T] = \left[\frac{M^tM}{\sigma^2} + \Phi^t[-1]\ H^t[nT]\left[R_{1t}[nT]\right]^{-1} H[nT]\ \Phi[-1]\right]^{-1} \tag{75}$$

Substitution of (51) into (75) and the use of the inverse of (56) gives

$$\hat{S}[(n+1)T] = \left[\frac{M^tM}{\sigma^2} + \left[S_1[nT]\right]^{-1}\right]^{-1} \tag{76}$$

Equation (76) along with (64) now forms a recursion for the covariance of the estimate.

For the quantities in (76) the application of the matrix inversion lemma [5], gives

$$\left[\frac{M^t M}{\sigma^2} + \left[S_1[nT]\right]^{-1}\right]^{-1} = S_1[nT] - S_1[nT]\, M^t[M\, S_1[nT]\, M^t + \sigma^2]^{-1}\, M\, S_1[nT] \tag{77}$$

This form will be used to generate an expression for the Kalman gain. Post multiplying both sides of (75) by $M^t/\sigma^2$ and using (75) the Kalman gain is defined as

$$\begin{aligned} K[(n+1)T] &= \hat{S}[(n+1)T]\, M^t/\sigma^2 \\ &= S_1[(n+1)T]\cdot M^t[\sigma^2 + M\, S_1[(n+1)T]\, M^t]^{-1} \end{aligned} \tag{78}$$

Thus the covariance matrix given by (76) becomes

$$\hat{S}[(n+1)T] = [I - K[(n+1)T]\, M]\, S_1[(n+1)T] \tag{79}$$

The recursion for the optimal estimate is now formed that uses the Kalman gain given by (78). Using the form for the optimal estimate given by (47) the estimate $\hat{\underline{x}}[(n+1)T]$ is

$$\hat{\underline{x}}[(n+1)T] = \hat{S}[(n+1)T]\, H^t[(n+1)T]\left[R_{1t}[(n+1)T]\right]^{-1} \underline{y}_t[(n+1)T]. \tag{80}$$

where $\underline{y}_t[(n+1)T]$ is the new total observation vector given by

$$\underline{y}_t[(n+1)T] = \begin{bmatrix} y[(n+1)T] \\ \hline \underline{y}_t\,[nT] \end{bmatrix} \tag{81}$$

Substitution of (74), (72) and (81) into (80) and carrying out the matrix multiplication yields

$$\hat{\underline{x}}[(n+1)T] = \hat{S}[(n+1)T]\left[\frac{M^t\, \underline{y}[(n+1)T]}{\sigma^2} + \Phi[-1]\, H^t[nT]\left[R_{1t}[nT]\right]^{-1} \underline{y}_t[nT]\right] \tag{82}$$

Using the estimate of the forecast $x_1[(n+1)T]$ (82) is simplified to

$$\hat{\underline{x}}[(n+1)T] = \hat{S}[(n+1)T]\left[\frac{M^t}{\sigma^2}\, y[(n+1)T] + \left[S_1[(n+1)T]\right]^{-1} x_1[(n+1)T]\right]. \quad (83)$$

Adding and subtracting $(M^t M/\sigma^2)\ x_1[(n+1)T]$, performing some algebraic manipulation, and applying (76) now yields

$$\underline{x}[(n+1)T] = \underline{x}_1[(n+1)T] + K[(n+1)T]\left[y[(n+1)T] - M\ \underline{x}_1[(n+1)T]\right] \quad (84)$$

where $K[(n+1)T]$ is given by (78). Equations (84), (79), (78), (64) and (62) now form the recursive formulation called the Kalman filter. These equations are now summarized as

$$\underline{x}_1(nT) = \Phi(1)\ \hat{\underline{x}}[(n-1)T]\ , \quad (85)$$

$$S_1(nT) = \Phi(1)\ \hat{S}[(n-1)T]\ \Phi^t(1) + Q\ , \quad (86)$$

$$K(nT) = S_1(nT)\ M^t[\sigma^2 + M\ S_1(nT)\ M^t]^{-1}\ , \quad (87)$$

$$\hat{S}(nT) = [I - K(nT)\ M]\ S_1(nT) \quad (88)$$

and

$$\begin{aligned}\hat{\underline{x}}(nT) &= \underline{x}_1(nT) + K(nT)\ [y(nT) - M\ \underline{x}_1(nT)] \\ &= [I - K(nT)M]\ \Phi(1)\ \hat{x}[(n-1)T] + K(nT)\ y(nT)\end{aligned} \quad (89)$$

In this set of equations, $x_1(nT)$ is the forecast or prediction of the estimate at $t=nT$ using the previously generated estimate at $t=[(n-1)T]$. The covariance of the forecast is $S_1(nT)$. The term $K(nT)$ is the time varying Kalman gain matrix and $y(nT)$ is the observation at $t=nT$. The terms $\hat{x}(nT)$ and $\hat{S}(nT)$ are the estimate at $t=nT$ and its covariance. The term $\sigma^2$ is the variance of the measurement noise and the term $Q$ is the covariance of the driving noise. It is the term $Q$ that serves as a key design parameter.

If there is no driving noise so that the diagonal elements of Q are all zero, the filter is simply an expanding memory filter. This means that if the filter is initialized properly, the estimates will correspond to those obtained by designing nonrecursive filters where the data window starts at zero and a new weight matrix $\hat{W}(nT)$ is computed as each new measurement is made. Obviously as the window expands, the variance of the estimate will decrease; however, if the model is not exact deterministic errors will begin to increase.

In using the equations, they must be initialized properly if a truly unbiased estimate is to be formed. In practice this is usually done using a nonrecursive filter. For exact initialization, driving noise must be included in the computation of the nonrecursive filter weights. To minimize the computation, the minimum H matrix should be used which means the number of rows should equal the number of columns. By using this minimum H matrix the filter weights are computed and the initial optimal estimate is formed from the actual data.

## USING THE PROGRAMS

### HARDWARE

The programs are written in DOS FORTRAN. They were developed on a PDP 11/20 with a DOS/BATCH operating system. Printed results are written to logical unit 5, which can be assigned at run time to a line printer, CRT terminal, disk file, or other suitable output device. Data is entered from units 6 and 3, which can be assigned to a card reader, disk data file, TTY keyboard, or any other suitable input device. Plots can be obtained with a GT40 graphics display terminal and the plotting subroutines provided.. The programs can be easily modified to use other plotting routines.

NONRECURSIVE FILTER PROGRAM

The program for generating the coefficients for non-recursive filters is called PROGRAM WINDOW. WINDOW is written in DEC FORTRAN, but may be run on other versions of FORTRAN IV with minor modifications. The program can call plotting packages to produce CRT or hardcopy plots of the filter response to various inputs.

Program WINDOW reads the filter parameters SIGMA, N, M, IA, IPLOT from logical unit 6, where:

| | |
|---|---|
| SIGMA | determines the input covariance matrix R. If SIGMA > 0.0, R becomes $\sigma^2 * I$, where $\sigma^2$ = SIGMA and I is the identity matrix. If SIGMA $\leq$ 0.0, the matrix R is read from unit 6 in 10F8.2 FORMAT. |
| N | is the number of points in the window ($1 \leq N \leq 20$). |
| M | is the order of the polynomial fit ($1 \leq M \leq 9$). |
| IA | is the offset $\alpha$ from the first point in the window. If IA > 0, the filter predicts IA sample times ahead of the most recent sample. If IA < 0, the filter smooths IA sample times behind the most recent sample. |
| IPLOT | is the plotting control variable. If IPLOT = 0, the program finishes after the coefficient matrices are computed and printed out. If IPLOT ≠ 0, an input signal is read, and the input points are filtered using the coefficients computed. |

The parameters are read in F10.4, 4I3 FORMAT.

Once the filter parameters have been read, WINDOW generates the required S (covariance) and T matrices, and computes the weight matrix W. All three matricies are then written to logical unit 5. If IPLOT = 0, the program terminates after the weight matrix W has been printed.

If IPLOT ≠ 0, WINDOW reads 101 sample points from logical unit 3 in G15.6 FORMAT. These points are provided by the user, and are used by WINDOW as the filter input signal. The program filters this input signal

using the weight matrix W, and tabulates the input signal, output signal (filter response), and error signal (input minus output) for each sample time. The tabulated results are printed on unit 5. The sum of the total absolute error is also computed and written to unit 5. WINDOW then calls the plotting package routines (subroutine IDIOT) to plot the input and output (estimate) signals vs. time. After a PAUSE, the program calls the plotting package to plot the error signal (input minus output) vs. time. The plotting packages are included to be used with WINDOW, or WINDOW may be changed to call other plotting routines.

Program WINDOW can be modified to write the derivative estimates. Note from equation 2 that the $m^{th}$ derivative term is multiplied by a $T^m/m!$ factor, where T is the sample time. Thus, if the derivative estimates are written out, they will be scaled by this factor. To illustrate the use of the program, WINDOW was run with the parameters SIGMA = 1.0, N = 5, M = 3, IA = 0, and IPLOT = 0. The filter weighting coefficients are listed row by row, and are shown with the input parameters and S and T matrices in Fig. 1. The filter obtained using the weight matrix shown estimates the input data and the first three derivatives. In equation form, these estimates are given by

$$\hat{x}[nT] = 0.9857\ y[nT] + 0.05714\ y[(n-1)T] - 0.0857\ y[(n-2)T] + 0.05714\ y[(n-3)T] - 0.01429\ y[(n-4)T]\ ,$$

$$\dot{\hat{x}}[nT] = \frac{1.488\ y[nT] - 1.619\ y[(n-1)T] - .5714\ y[(n-2)T]}{T} + \frac{1.048\ y[(n-3)T] - 0.3452\ y[(n-4)T]}{T}\ ,$$

$$\ddot{\hat{x}}[nT] = \frac{0.6429\ y[nT] - 1.071\ y[(n-1)T] - 0.1429\ y[(n-2)T]}{T^2/2} + \frac{0.9286\ y[(n-3)T] - 0.3571\ y[(n-4)T]}{T^2/2}\ ,$$

$$\dddot{\hat{x}}[nT] = \frac{0.08333\ y[nT] - 0.1667\ y[(n-1)T]}{T^3/6}$$

$$+ \frac{0.1667\ y[(n-3)T] - 0.08333\ y[(n-4)T]}{T^3/6}.$$

RECURSIVE FILTERS

The Kalman filter program is called ADAPT. ADAPT is written in DEC FORTRAN, but may be run on other versions of FORTRAN IV with minor modifications. The program can call plotting packages to produce CRT or hardcopy plots of the filter response to various inputs.

Program ADAPT reads the filter parameters SIGMA, M, and Q from logical unit 6, where

| | |
|---|---|
| SIGMA | determines the initial covariance matrix R. If SIGMA > 0.0, R becomes $\sigma^2$*I, where $\sigma^2$ = SIGMA and I is the identity matrix. If SIGMA $\leq$ 0.0, the matrix R is read from unit 6 in 10F8.2 FORMAT. |
| M | is the order of the polynomial fit ($1 \leq M \leq 9$). |
| Q | is the driving noise term. |

The parameters are read in F10.4,I3,F10.4 FORMAT.

Once the filter parameters have been read, ADAPT generates an initial S (covariance) and T matrix. ADAPT then generates an initial weight matrix W, which is used to initialize the $\underline{x}$ vector. The initial covariance matrix is used to initialize the S matrix.

Once initialized, ADAPT reads 101 sample points from logical unit 3 in G15.6 FORMAT. These points are provided by the user, and are used by ADAPT as the Kalman filter input. The program filters the input using equations 85 through 89. The sample number filter input, filter output, error signal (input minus output), and Kalman gain are tabulated and written to unit 5. The total absolute error is also computed and written

to unit 5. ADAPT then calls the plotting package routines (subroutine IDIOT) to plot the input and output (estimate) signals vs. time. After a PAUSE, the program calls the plotting package to plot the error signal (input minus output) vs. time. After a second PAUSE, IDIOT is called to plot the Kalman gain vs. time.

Program ADAPT can be modified to write the derivative estimates. Note from equation 2 that the $m^{th}$ derivative term is multiplied by a $T^m/m!$ factor, where T is the sample time. Thus, if the derivative estimates are written out, they will be scaled by this factor.

To illustrate the use of the program, ADAPT was run with the parameters SIGMA = 1.0, M = 3, Q = 0.0. The S (covariance) and T matrices used to generate the initial weight matrix (W) are shown in Fig. 2. The S and W matrices are used to initialize the Kalman filters' $\hat{S}$ and $\underline{\hat{x}}$ matrices. The Kalman filter was then used to filter an ideal sinusoidal signal. A portion of the tabulated results is shown in Fig. 3.

GETTING ON LINE

In order to run program WINDOW and ADAPT, first build two files named WINDOW.FTN and ADAPT.FTN from the sources provided (card deck or paper tape). Also create PLT.FTN and SENDGT.MAC from the sources. Compile programs WINDOW and ADAPT with the FORTRAN compiler to create WINDOW.OBJ and ADAPT.OBJ. The one word integer option should be selected for all compilations. Also, compile PLT.FTN to create PLT.OBJ. Assemble SENDGT.MAC under the MACRO assembler to create SENDGT.OBJ. Create a subroutine library called PLTLIB.OBJ from PLT.OBJ and SENDGT.OBJ (in that order). Next, LINK WINDOW.OBJ, PLTLIB.OBJ and the FTN library to create a file called WINDOW.LDA.

Also, create ADAPT.LDA by LINKing ADAPT.OBJ, PLTLIB.OBJ and the FTN library. The files WINDOW.OBJ, ADAPT.OBJ, PLT.OBJ and SENDGT.OBJ may now be deleted.

Before running WINDOW or ADAPT, build the file PLOTGT.MAC from the source. PLOTGT is the plotting routine that is loaded into the GT40. Assemble and LINK PLOTGT so that PLOTGT.LDA may be loaded into the GT40. After PLOTGT.LDA is running in the GT40, ADAPT.LDA or WINDOW.LDA may be executed using a RUN command. The source listings contain additional documentation on these programs.

***FINITE MEMORY DIGITAL FILTER PACKAGE***

```
VARIANCE OF ERROR=     1.0000
SIZE OF WINDOW=  5
ORDER OF FIT=  3
TO PREDICT   0 UNITS AWAY FROM THE FRONT OF THE WINDOW


        MAGIC T MATRIX
1.0000        0.0000         0.0000         0.0000
1.0000       -1.0000         1.0000        -1.0000
1.0000       -2.000          4.000         -8.000
1.0000       -3.000          9.000         -27.00
1.0000       -4.000          16.00         -64.00


        VARIANCE MATRIX, S
0.9857        1.488          0.6429         0.8333E-01
 1.488        6.279          3.869          0.5972
0.6429        3.869          2.571          0.4167
0.8333E-01    0.5972         0.4167         0.6944E-01


       WINDOW WEIGHTS,  FROM FRONT TO BACK

              ROW  1 OF W,  CORRESPONDING TO THE INPUT SIGNAL
0.9857        0.5714E-01  -0.8571E-01   0.5714E-01  -0.1429E-01

              ROW  2 OF W,  CORRESPONDING TO DERIVATIVE NUMBER  1
 1.488       -1.619       -0.5714        1.048        -0.3452

              ROW  3 OF W,  CORRESPONDING TO DERIVATIVE NUMBER  2
0.6429       -1.071       -0.1429        0.9286       -0.3571

              ROW  4 OF W,  CORRESPONDING TO DERIVATIVE NUMBER  3
0.8333E-01   -0.1667      -0.3576E-06    0.1667       -0.8333E-01
```

Fig. 1 - Program WINDOW Sample Output

***KALMAN STRUCTURE DIGITAL FILTER PACKAGE***

DRIVING NOISE= 0.0000
ORDER OF FIT= 3
INITIAL ERROR VARIANCE 1.0000

MAGIC T MATRIX

| | | | |
|---|---|---|---|
| 1.0000 | 0.0000 | 0.0000 | 0.0000 |
| 1.0000 | -1.0000 | 1.0000 | -1.0000 |
| 1.0000 | -2.000 | 4.000 | -8.000 |
| 1.0000 | -3.000 | 9.000 | -27.00 |

VARIANCE MATRIX, S

| | | | |
|---|---|---|---|
| 1.0000 | 1.833 | 1.0000 | 0.1667 |
| 1.833 | 14.72 | 12.50 | 2.611 |
| 1.0000 | 12.50 | 11.50 | 2.500 |
| 0.1667 | 2.611 | 2.500 | 0.5555 |

WINDOW WEIGHTS, FROM FRONT TO BACK

ROW 1 OF W, CORRESPONDING TO THE INPUT SIGNAL

1.0000 0.4128E-05 -0.2503E-05 0.9537E-06

ROW 2 OF W, CORRESPONDING TO DERIVATIVE NUMBER 1

1.833 -3.000 1.500 -0.3333

ROW 3 OF W, CORRESPONDING TO DERIVATIVE NUMBER 2

1.0000 -2.500 2.000 -0.5000

ROW 4 OF W, CORRESPONDING TO DERIVATIVE NUMBER 3

0.1667 -0.5000 0.5000 -0.1667

Fig. 2 - Program ADAPT Sample Output

| TIME | OUTPUT | ESTIMATE | ERROR | KAL GAIN |
|---|---|---|---|---|
| 4.000 | 0.3894 | 0.3894 | 0.2682E-06 | 0.9857 |
| 5.000 | 0.4794 | 0.4794 | 0.1627E-04 | 0.9695 |
| 6.000 | 0.5646 | 0.5646 | 0.5460E-04 | 0.9482 |
| 7.000 | 0.6442 | 0.6441 | 0.1017E-03 | 0.9162 |
| 8.000 | 0.7174 | 0.7172 | 0.1562E-03 | 0.8794 |
| 9.000 | 0.7833 | 0.7831 | 0.2229E-03 | 0.8419 |
| 10.000 | 0.8415 | 0.8412 | 0.3122E-03 | 0.8053 |
| 11.00 | 0.8912 | 0.8908 | 0.4330E-03 | 0.7707 |
| 12.00 | 0.9320 | 0.9314 | 0.6002E-03 | 0.7382 |
| 13.00 | 0.9636 | 0.9627 | 0.8295E-03 | 0.7079 |
| 14.00 | 0.9855 | 0.9843 | 0.1140E-02 | 0.6797 |
| 15.00 | 0.9975 | 0.9959 | 0.1551E-02 | 0.6534 |
| 16.00 | 0.9996 | 0.9975 | 0.2086E-02 | 0.6289 |
| 17.00 | 0.9917 | 0.9889 | 0.2768E-02 | 0.6061 |
| 18.00 | 0.9738 | 0.9702 | 0.3620E-02 | 0.5848 |
| 19.00 | 0.9463 | 0.9416 | 0.4669E-02 | 0.5649 |
| 20.00 | 0.9093 | 0.9034 | 0.5938E-02 | 0.5462 |
| 21.00 | 0.8632 | 0.8558 | 0.7450E-02 | 0.5287 |
| 22.00 | 0.8085 | 0.7993 | 0.9229E-02 | 0.5123 |
| 23.00 | 0.7457 | 0.7344 | 0.1130E-01 | 0.4968 |
| 24.00 | 0.6755 | 0.6618 | 0.1367E-01 | 0.4822 |
| 25.00 | 0.5985 | 0.5821 | 0.1635E-01 | 0.4684 |
| 26.00 | 0.5155 | 0.4961 | 0.1937E-01 | 0.4553 |
| 27.00 | 0.4274 | 0.4047 | 0.2272E-01 | 0.4430 |
| 28.00 | 0.3350 | 0.3086 | 0.2640E-01 | 0.4312 |
| 29.00 | 0.2392 | 0.2088 | 0.3040E-01 | 0.4201 |
| 30.00 | 0.1411 | 0.1064 | 0.3472E-01 | 0.4095 |
| 31.00 | 0.4158E-01 | 0.2265E-02 | 0.3932E-01 | 0.3995 |
| 32.00 | -0.5837E-01 | -0.1025 | 0.4[illegible]E-01 | 0.3899 |
| 33.00 | -0.1577 | -0.2070 | [illegible]E-01 | 0.3807 |
| 34.00 | -0.2555 | -0.3100 | [illegible]E-01 | 0.3720 |
| 35.00 | -0.3508 | -0.4106 | [illegible]981E-01 | 0.3637 |
| 36.00 | -0.4425 | -0.5077 | 0.6519E-01 | 0.3557 |
| 37.00 | -0.5298 | -0.6004 | 0.7054E-01 | 0.3480 |
| 38.00 | -0.6119 | -0.6876 | 0.7576E-01 | 0.3407 |
| 39.00 | -0.6878 | -0.7685 | 0.8076E-01 | 0.3337 |
| 40.00 | -0.7568 | -0.8422 | 0.8543E-01 | 0.3269 |
| 41.00 | -0.8183 | -0.9080 | 0.8968E-01 | 0.3204 |
| [illegible] 00 | -0.8716 | -0.9649 | 0.9336E-01 | 0.3142 |
| 43.00 | -0.9162 | -1.013 | 0.9637E-01 | 0.3082 |
| 44.00 | -0.9516 | -1.050 | 0.9857E-01 | 0.3024 |
| 45.00 | -0.9775 | -1.077 | 0.9981E-01 | 0.2969 |
| 46.00 | -0.9937 | -1.094 | 0.9997E-01 | 0.2915 |
| 47.00 | -0.9999 | -1.099 | 0.9891E-01 | 0.2863 |
| 48.00 | -0.9962 | -1.093 | 0.9648E-01 | 0.2813 |
| 49.00 | -0.9825 | -1.075 | 0.9255E-01 | 0.2765 |
| 50.00 | -0.9589 | -1.046 | 0.8697E-01 | 0.2718 |
| 51.00 | -0.9258 | -1.005 | 0.7963E-01 | 0.2673 |
| 52.00 | -0.8835 | -0.9538 | 0.7039E-01 | 0.2629 |
| 53.00 | -0.8323 | -0.8914 | 0.5915E-01 | 0.2587 |
| 54.00 | -0.7728 | -0.8186 | 0.4580E-01 | 0.2546 |
| 55.00 | -0.7055 | -0.7358 | 0.3025E-01 | 0.2507 |
| 56.00 | -0.6313 | -0.6437 | 0.1241E-01 | 0.2468 |
| 57.00 | -0.5507 | -0.5429 | -0.7760E-02 | 0.2431 |
| 58.00 | -0.4646 | -0.4343 | -0.3032E-01 | 0.2395 |
| 59.00 | -0.3739 | -0.3186 | -0.5528E-01 | 0.2360 |
| 60.00 | -0.2794 | -0.1968 | -0.8265E-01 | 0.2325 |
| 61.00 | -0.1822 | -0.6975E-01 | -0.1124 | 0.2292 |

Fig. 3 - Program ADAPT Sample Output

| | | | | |
|---|---|---|---|---|
| 0.9857 | 0.05714 | -0.0857 | 0.05714 | -0.01429 |
| 1.488 | -1.619 | -0.5714 | 1.048 | -0.3452 |
| 0.6429 | -1.071 | -0.1429 | 0.9286 | -0.3571 |
| 0.08333 | -0.1667 | 0.0 | 0.1667 | -0.08333 |

Weight Matrix

| | | | |
|---|---|---|---|
| 0.9857 | 1.488 | 0.6429 | 0.08333 |
| 1.488 | 6.379 | 3.869 | 0.5972 |
| 0.6429 | 3.869 | 2.571 | 0.4167 |
| 0.08333 | 0.5972 | 0.4167 | 0.06944 |

Covariance Matrix

Table 1 Optimal Weight Matrix and Covariance Matrix For a Five Point Nonrecursive Filter Using a Third Order Polynomial Model. The Filter Estimates the Data at the End of the Window

$$\begin{bmatrix} -0.08571 & 0.3429 & 0.4857 & 0.3429 & -0.08571 \\ -0.08333 & 0.6667 & 0.0 & -0.6667 & 0.08333 \\ 0.1429 & -0.07143 & -0.1429 & -0.07143 & 0.1429 \\ 0.08333 & -0.1667 & 0.0 & 0.1667 & -0.08333 \end{bmatrix}$$

Weight Matrix

$$\begin{bmatrix} 0.4857 & 0.0 & -0.1429 & 0.0 \\ 0.0 & 0.9828 & 0.0 & -0.2361 \\ -0.1429 & 0.0 & 0.07143 & 0.0 \\ 0.0 & -0.2361 & 0.0 & 0.06944 \end{bmatrix}$$

Covariance Matrix

Table 2 Optimal Weight Matrix and Covariance Matrix For a Five Point Nonrecursive Filter Using a Third Order Polynomial Model. The Filter Estimates the Data in the Center of the Window

## REFERENCES


1. P. DeRusso, R. Roy and C. Close, State Variables for Engineers, John Wiley and Sons, 1965.

2. C. Chen, Introduction to Linear System Theory, Holt, Rinehart and Winston, 1970.

3. A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," Analytical Chemistry, Vol. 36, No. 8, pp. 1627-1639, July 1964.

4. N. Morrison, Introduction to Sequential Smoothing and Prediction, McGraw-Hill, 1969.

5. A. Sage and J. Melso, Estimation Theory with Applications to Communications and Control, McGraw-Hill, 1971.